

Universidad Carlos III de Madrid

Escuela Politécnica Superior



**Ingeniería Técnica en Informática de
Gestión**

Proyecto Fin de Carrera

**“Creación de la Aplicación "¿Quién es quién?" para
Facebook”**

Autor: Ruiz Gualda, Francisco

Tutor: Rivero Espinosa, Jessica

Noviembre, 2011

Agradecimientos



Índice de Figuras

Figura 1. Ejemplos de redes sociales, una pequeña descripción	20
Figura 2. Imagen del juego Farmville	22
Figura 3. Capturas de la aplicación Dropio.io	22
Figura 4. Juego de mafia Mob Wars	22
Figura 5. KaChing	23
Figura 6. Aplicación de YouTube para Facebook	23
Figura 7. Mapa con las ciudades que has visitado	23
Figura 8. Página principal de usuario de Facebook Developer	28
Figura 9. Aceptar condiciones de Facebook	28
Figura 10. Página principal de configuración de la aplicación	29
Figura 11. Miarroba.com ofrece hosting gratuito	29
Figura 12. Información general sobre el hosting	30
Figura 13. Gestor de archivos de miarroba	30
Figura 14. Editor de ficheros de miarroba.com	31
Figura 15. Apartado Lienzo de la configuración de la aplicación	31
Figura 16. Página de la aplicación, con información y estadísticas de esta	32
Figura 17. Detalle del apartado Connect	32
Figura 18. Resultado final	33
Figura 19. Detalle Flash Builder 4	35
Figura 20. Seleccionar Desktop. Pestaña avanzada	36
Figura 21. Proyecto Web nuevo	36
Figura 22. Package Explorer de AFB4	37
Figura 23. Vista general AFB4	37
Figura 24. Vista diseño AFB4	38
Figura 25. Ventana propiedades de cada lista insertada	38
Figura 26. Lista álbumes	39
Figura 27. Lista fotos	39
Figura 28. Código generado automáticamente	40
Figura 29. Bloque Script	40
Figura 30. Clases y variables de la aplicación	41
Figura 31. Función init()	41
Figura 32. Función deLogin()	42
Figura 33. Función handleLogin()	42
Figura 34. Función getPhotoAlbums	42
Figura 35. Función handleGetAlbumsResponse()	43
Figura 36. Función getPhotoAlbums()	43
Figura 37. Función handleGetPhotosResponse()	43
Figura 38. Sentencia perteneciente a la lista photoList	43
Figura 39. Login en Facebook	44
Figura 40. Visualización de los álbumes y sus miniaturas	45
Figura 41. Diagrama casos de uso	54
Figura 42. Ciclo de vida. Modelo de prototipo	81
Figura 43. RBS	83
Figura 44. Proceso WBS. Estudio de Viabilidad	84
Figura 45. Proceso WBS. Gestión del Proyecto	85
Figura 46. Proceso WBS. Primer Prototipo	86
Figura 47. Proceso WBS. Segundo Prototipo	87
Figura 48. Proceso WBS. Tercer Prototipo	88



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

Figura 49. Proceso WBS. Manual de Usuario	89
Figura 50. Diagrama de Gantt. General	90
Figura 51. Diagrama de Gantt. Estudio de Viabilidad	90
Figura 52. Diagrama de Gantt. Gestión del Proyecto	91
Figura 53. Diagrama de Gantt. Primer prototipo	92
Figura 54. Diagrama de Gantt. Segundo prototipo	92
Figura 55. Diagrama de Gantt. Tercer prototipo	93
Figura 56. Diagrama de Gantt. Manual de usuario	93
Figura 57. Diagrama de Gantt. Corrección de manuales y entrega	94
Figura 58. Diagrama de secuencia. Caso de uso CU-001. Iniciar juego	127
Figura 59. Diagrama de secuencia. Caso de uso CU-002. Cambiar vista	128
Figura 60. Diagrama de secuencia. Caso de uso CU-003. Ver instrucciones	128
Figura 61. Diagrama de secuencia. Caso de uso CU-004. Realizar pregunta a Com	129
Figura 62. Diagrama de secuencia. Caso de uso CU-005. Contestar	129
Figura 63. Diagrama de secuencia. Caso de uso CU-006. Descartar	130
Figura 64. Diagrama de secuencia. Caso de uso CU-008. Resolver Usuario	130
Figura 65. Diagrama de secuencia. Caso de uso CU-008. Resolver Usuario	131
Figura 66. Diagrama de secuencia. Caso de uso CU-014. Realizar pregunta a	131
Figura 67. Diagrama de secuencia. Caso de uso CU-016. Realizar pregunta a	132
Figura 68. Diagrama conceptual de clases	133
Figura 69. Interfaz de la aplicación	138
Figura 70. Interfaz Usuario	140
Figura 71. Interfaz Usuario. Parte superior	140
Figura 72. Interfaz Usuario. Objetivo Com y visualizador de detalles	141
Figura 73. Interfaz Usuario. Objetivo Com y visualizador de detalles	141
Figura 74. Interfaz Usuario. Objetivo Com y visualizador de detalles	142
Figura 75. Interfaz Usuario. Objetivo Com y visualizador de detalles	142
Figura 76. Interfaz Usuario. Controles del juego	142
Figura 77. Interfaz Usuario. Controles del juego. Turno Usuario	143
Figura 78. Interfaz Usuario. Controles del juego. Turno Usuario	143
Figura 79. Interfaz Usuario. Controles del juego. Turno Usuario. Etapa ceder	144
Figura 80. Interfaz Usuario. Controles del juego. Turno Usuario. Etapa resolver	144
Figura 81. Interfaz Usuario. Controles del juego. Turno Usuario. Etapa	144
Figura 82. Interfaz Usuario. Controles del juego. Turno Com	145
Figura 83. Interfaz Com	146
Figura 84. Interfaz Com. Controles superiores	146
Figura 85. Interfaz Com. Tablero Com	147
Figura 86. Interfaz Com. Objetivo Usuario y visualizador de detalles	147
Figura 87. Interfaz Com. Objetivo Usuario y visualizador de detalles	148
Figura 88. Interfaz Com. Objetivo Usuario y visualizador de detalles	148
Figura 89. Cabeceras de las clases	154
Figura 90. Componentes de la arquitectura del sistema	156
Figura 91. Diagrama de secuencia extendido. Caso de uso CU-001. Iniciar juego	159
Figura 92. Diagrama de secuencia extendido. Caso de uso CU-008	160
Figura 93. Diagrama de clases de diseño	161
Figura 94. Arquitectura de una aplicación en Facebook.com	167
Figura 95. Interfaz Aplicación funcionando bajo el iFrame de Facebook	189
Figura 96. Interfaz Usuario funcionando bajo el iFrame de Facebook	190
Figura 97. Interfaz Com funcionando bajo el iFrame de Facebook	191
Figura 98. Interfaz Instrucciones	192



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

Figura 99. Realización de la prueba PR-001	196
Figura 100. Realización de la prueba PR-002	197
Figura 101. Realización de la prueba PR-003	198
Figura 102. Realización de la prueba PR-004	199
Figura 103. Realización de la prueba PR-005	200
Figura 104. Realización de la prueba PR-006	201
Figura 105. Realización de la prueba PR-007	202
Figura 106. Realización de la prueba PR-008	203
Figura 107. Realización de la prueba PR-009	204
Figura 108. Realización de la prueba PR-010	205
Figura 109. Realización de la prueba PR-011	206
Figura 110. Realización de la prueba PR-012	207
Figura 111. Realización de la prueba PR-013	208
Figura 112. Realización de la prueba PR-014	209
Figura 113. Realización de la prueba PR-015	210
Figura 114. Realización de la prueba PR-016	211
Figura 115. Realización de la prueba PR-017. Pregunta lanzada por Com	212
Figura 116. Realización de la prueba PR-017. Situación del tablero de Com	213
Figura 117. Realización de la prueba PR-018. Interfaz Usuario Resolver	214
Figura 118. Realización de la prueba PR-018. InterfazCom, objetivo Usuario	215
Figura 119. Realización de la prueba PR-019. InterfazUsuario	216
Figura 120. Realización de la prueba PR-019. InterfazCom, Comprobando tablero	216
Figura 121. Realización de la prueba PR-020. Desconexión de Facebook	217
Figura 122. Realización de la prueba PR-021. Descarte Realizado, ceder turno	218
Figura 123. Realización de la prueba PR-021. Turno cedido, pregunta Com	218
Figura 124. Gráfico Valoración final / Edad	224
Figura 125. Iniciar sesión Facebook	230
Figura 126. Empezar juego	231
Figura 127. Vista general de la aplicación	231
Figura 128. Controles del juego. Pregunta	232
Figura 129. Controles del juego. Respuesta de oponente a pregunta	232
Figura 130. Controles del juego. Respuesta de usuario a pregunta del oponente	233
Figura 131. Controles del juego. Resolver	233



Índice de Tablas

Tabla 1. Actor Usuario	55
Tabla 2. Actor Com	55
Tabla 3. Caso de uso CU-001	56
Tabla 4. Caso de uso CU-002	57
Tabla 5. Caso de uso CU-003	57
Tabla 6. Caso de uso CU-004	58
Tabla 7. Caso de uso CU-005	59
Tabla 8. Caso de uso CU-006	60
Tabla 9. Caso de uso CU-007	60
Tabla 10. Caso de uso CU-008	61
Tabla 11. Caso de uso CU-009	61
Tabla 12. Caso de uso CU-010	62
Tabla 13. Caso de uso CU-011	62
Tabla 14. Caso de uso CU-012	63
Tabla 15. Caso de uso CU-013	63
Tabla 16. Caso de uso CU-014	64
Tabla 17. Caso de uso CU-015	65
Tabla 18. Caso de uso CU-016	66
Tabla 19. Caso de uso CU-017	66
Tabla 20. Requisito de usuario RU-C-001	69
Tabla 21. Requisito de usuario RU-C-002	69
Tabla 22. Requisito de usuario RU-C-003	70
Tabla 23. Requisito de usuario RU-C-004	70
Tabla 24. Requisito de usuario RU-C-005	71
Tabla 25. Requisito de usuario RU-C-006	72
Tabla 26. Requisito de usuario RU-C-007	73
Tabla 27. Requisito de usuario RU-C-008	73
Tabla 28. Requisito de usuario RU-C-009	74
Tabla 29. Requisito de usuario RU-C-010	74
Tabla 30. Requisito de usuario RU-C-011	74
Tabla 31. Requisito de usuario RU-C-012	75
Tabla 32. Requisito de usuario RU-C-013	75
Tabla 33. Requisito de usuario RU-C-014	75
Tabla 34. Requisito de usuario RU-R-001	76
Tabla 35. Requisito de usuario RU-R-002	76
Tabla 36. Requisito de usuario RU-R-003	77
Tabla 37. Requisito de usuario RU-R-004	77
Tabla 38. Requisito de usuario RU-R-005	77
Tabla 39. Requisito de usuario RU-R-006	78
Tabla 40. Requisito de usuario RU-R-007	78
Tabla 41. Requisito de usuario RU-R-008	78
Tabla 42. Requisito de usuario RU-R-009	79
Tabla 43. Requisito de usuario RU-R-010	79
Tabla 44. Costes de licencias	95
Tabla 45. Costes de recursos humanos	96
Tabla 46. Coste total del proyecto	96
Tabla 47. Requisito software RS-F-001	103
Tabla 48. Requisito software RS-F-002	103



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

Tabla 49. Requisito software RS-F-003	104
Tabla 50. Requisito software RS-F-004	105
Tabla 51. Requisito software RS-F-005	105
Tabla 52. Requisito software RS-F-006	106
Tabla 53. Requisito software RS-F-007	106
Tabla 54. Requisito software RS-F-008	107
Tabla 55. Requisito software RS-F-009	108
Tabla 56. Requisito software RS-F-010	109
Tabla 57. Requisito software RS-F-011	109
Tabla 58. Requisito software RS-F-012	110
Tabla 59. Requisito software RS-F-013	111
Tabla 60. Requisito software RS-F-014	112
Tabla 61. Requisito software RS-F-015	112
Tabla 62. Requisito software RS-F-016	113
Tabla 63. Requisito software RS-F-017	113
Tabla 64. Requisito software RS-F-018	114
Tabla 65. Requisito software RS-F-019	114
Tabla 66. Requisito software RS-F-020	115
Tabla 67. Requisito software RS-F-021	115
Tabla 68. Requisito software RS-F-022	116
Tabla 69. Requisito software RS-F-023	116
Tabla 70. Requisito de usuario RU-I-001	117
Tabla 71. Requisito software RS-O-001	118
Tabla 72. Requisito software RS-O-002	118
Tabla 73. Requisito software RS-O-003	119
Tabla 74. Requisito software RS-O-004	119
Tabla 75. Requisito software RS-O-005	120
Tabla 76. Requisito software RS-O-006	121
Tabla 77. Requisito software RS-O-007	121
Tabla 78. Requisito software RS-O-008	122
Tabla 79. Requisito software RS-O-009	122
Tabla 80. Requisito software RS-O-010	123
Tabla 81. Requisito software RS-D-001	124
Tabla 82. Requisito software RS-C-001	125
Tabla 83. Requisito software RS-C-002	125
Tabla 84. Clases asociadas a casos de uso	126
Tabla 85. Clase Aplicación	134
Tabla 86. Clase InterfazUsuario	134
Tabla 87. Clase InterfazCom	135
Tabla 88. Clase Instrucciones	135
Tabla 89. Asociación clases Aplicación e InterfazUsuario	136
Tabla 90. Asociación clases Aplicación e InterfazCom	136
Tabla 91. Asociación clases InterfazUsuario e Instrucciones	136
Tabla 92. Matriz de trazabilidad Requisitos de Usuario – Casos de Uso	149
Tabla 93. Matriz de trazabilidad Requisitos software– Casos de Uso	150
Tabla 94. Componente C-001: Sistema	157
Tabla 95. Componente C-002: Facebook	157
Tabla 96. Clases de diseño asociadas a casos de uso	158
Tabla 97. Atributos, funciones y procedimientos clase Aplicación	162
Tabla 98. Atributos, funciones y procedimientos clase Aplicación	163



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

Tabla 99. Atributos, funciones y procedimientos clase Aplicación	166
Tabla 100. Matriz de trazabilidad Clases de diseño – Casos de uso	168
Tabla 101. Requisito software RS-IMP-01	170
Tabla 102. Prueba PR-001	177
Tabla 103. Prueba PR-002	177
Tabla 104. Prueba PR-003	178
Tabla 105. Prueba PR-004	178
Tabla 106. Prueba PR-005	178
Tabla 107. Prueba PR-006	179
Tabla 108. Prueba PR-007	179
Tabla 109. Prueba PR-008	180
Tabla 110. Prueba PR-009	180
Tabla 111. Prueba PR-010	181
Tabla 112. Prueba PR-011	181
Tabla 113. Prueba PR-012	181
Tabla 114. Prueba PR-013	182
Tabla 115. Prueba PR-014	182
Tabla 116. Prueba PR-015	182
Tabla 117. Prueba PR-016	183
Tabla 118. Prueba PR-017	183
Tabla 119. Prueba PR-018	184
Tabla 120. Prueba PR-019	184
Tabla 121. Prueba PR-020	184
Tabla 122. Prueba PR-021	185
Tabla 123. Plantilla informe de pruebas	186
Tabla 124. Matriz de trazabilidad Pruebas/Requisitos de capacidad	187
Tabla 125. Plantilla prueba PR-001	196
Tabla 126. Plantilla prueba PR-002	197
Tabla 127. Plantilla prueba PR-003	198
Tabla 128. Plantilla prueba PR-004	199
Tabla 129. Plantilla prueba PR-005	200
Tabla 130. Plantilla prueba PR-006	201
Tabla 131. Plantilla prueba PR-007	202
Tabla 132. Plantilla prueba PR-008	203
Tabla 133. Plantilla prueba PR-009	204
Tabla 134. Plantilla prueba PR-010	205
Tabla 135. Plantilla prueba PR-011	206
Tabla 136. Plantilla prueba PR-012	207
Tabla 137. Plantilla prueba PR-013	208
Tabla 138. Plantilla prueba PR-014	209
Tabla 139. Plantilla prueba PR-015	210
Tabla 140. Plantilla prueba PR-016	211
Tabla 141. Plantilla prueba PR-017	212
Tabla 142. Plantilla prueba PR-018	214
Tabla 143. Plantilla prueba PR-019	215
Tabla 144. Plantilla prueba PR-020	217
Tabla 145. Plantilla prueba PR-021	218
Tabla 146. Test de valoración. Sujeto 1	219
Tabla 147. Test de valoración. Sujeto 2	220
Tabla 148. Test de valoración. Sujeto 3	220



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

Tabla 149. Test de valoración. Sujeto 4	220
Tabla 150. Test de valoración. Sujeto 5	221
Tabla 151. Test de valoración. Sujeto 6	221
Tabla 152. Test de valoración. Sujeto 7	221
Tabla 153. Test de valoración. Sujeto 8	222
Tabla 154. Test de valoración. Sujeto 9	222
Tabla 155. Test de valoración. Sujeto 10	222
Tabla 156. Media total	223
Tabla 157. Media total Mujeres	223
Tabla 158. Media total Varones	223



Índice de contenidos

1. INTRODUCCIÓN.....	13
1.1 CONTEXTO.....	13
1.2 OBJETIVOS.....	14
1.3 ESTRUCTURA DEL DOCUMENTO	14
2. ESTADO DEL ARTE.....	16
2.1 INTRODUCCIÓN.....	16
2.2 INTRODUCCIÓN A LAS REDES SOCIALES.....	17
2.3 FACEBOOK.....	22
2.4 FACEBOOK DEVELOPERS.....	22
2.5 PERFIL DEL USUARIO DE LOS SOCIAL GAMES	25
2.6 DESARROLLO DE APLICACIONES CON FACEBOOK DEVELOPERS	26
2.6.1 Creando aplicaciones en Facebook	26
2.6.2 Estructura de una aplicación en Facebook.....	26
2.7 INTRODUCCIÓN A DEVELOPERS. PRUEBA.....	28
2.8 BÚSQUEDA DE TECNOLOGÍA	34
2.8.1 Creación de aplicaciones en Adobe Flash Builder 4 para Facebook.....	36
2.8.2 Funcionamiento del Programa	44
2.11 CONCLUSIONES.....	46
3. ESTUDIO DE VIABILIDAD DEL SISTEMA	47
3.1 INTRODUCCIÓN.....	47
3.1.2 Propósito del plan.	47
3.2 ESTABLECIMIENTO DEL ALCANCE DEL SISTEMA	48
3.2.1 Estudio de la solicitud.....	48
3.2.2 Identificación del alcance del sistema	50
3.2.3 Identificación de los interesados en el sistema (stakeholders)	51
3.3 ESTUDIO DE LA SITUACIÓN ACTUAL	52
3.3.1 Valoración del estudio de la situación actual	52
3.3.2 Realización del diagnóstico actual	53
3.4 DEFINICIÓN DE LOS REQUISITOS DEL SISTEMA	54
3.4.1 Especificación de los casos de uso	54
3.4.1.1 Especificación detallada de los actores.....	55
3.4.1.2 Especificación detallada de los casos de uso	56
3.4.2 Definición de los requisitos del sistema	67
3.4.2.1 Identificación de los requisitos.....	67
3.4.2.2 Requisitos de capacidad	69
3.4.2.3 Requisitos de restricción.....	76
3.5 ESTUDIO DE ALTERNATIVAS DE SOLUCIÓN	80
4. GESTIÓN DEL PROYECTO	81
4.1 CICLO DE VIDA.....	81
4.2 ORGANIZACIÓN DEL PROYECTO.....	83
4.3 PLANIFICACIÓN	90
4.4 ESTIMACIÓN DE COSTES	95
5. ANÁLISIS DEL SISTEMA.....	97
5.1 INTRODUCCIÓN.....	97
5.1.1 Objetivo del Análisis del Sistema.....	97
5.1.2 Alcance	97
5.2 DEFINICIÓN DEL SISTEMA	98
5.2.1 Determinación del alcance del sistema	98
5.2.2 Identificación del entorno tecnológico	98
5.2.3 Especificación de estándares y normas.....	98
5.2.3.1 Restricciones generales	99
5.2.3.2 Supuesto y dependencias	99
5.2.3.3 Entorno operacional	99



5.2.3.4	Identificación de los usuarios.....	100
5.2.3.5	Estudio de la seguridad requerida en el proceso de análisis.....	100
5.3	ESTABLECIMIENTO DE REQUISITOS DE SOFTWARE.....	101
5.3.1	Obtención de requisitos	101
5.3.1.1	Requisitos Funcionales	103
5.3.1.2	Requisitos de Interfaz.....	117
5.3.1.3	Requisitos de operación.....	118
5.3.1.4	Requisitos de documentación	124
5.3.1.5	Requisitos de calidad.....	125
5.4	ANÁLISIS DE LOS CASOS DE USO.....	126
5.4.1	Identificación de las clases asociadas a un caso de uso	126
5.4.2	Descripción de la interacción de objetos	127
5.5	ANÁLISIS DE CLASES.....	133
5.5.1	Identificación de responsabilidades y atributos	134
5.5.2	Identificación de asociaciones.....	136
5.6	DEFINICIÓN DE LAS INTERFACES DE USUARIO	137
5.6.1	Especificación de principios generales de la interfaz.....	137
5.6.2	Identificación de perfiles y diálogos.....	137
5.6.3	Especificación de formatos individuales de la interfaz de la pantalla	137
5.6.3.1	Interfaz de la aplicación	138
5.6.3.2	Interfaz Usuario	140
5.6.3.3	Interfaz Com.....	146
5.7	ANÁLISIS DE CONSISTENCIA Y ESPECIFICACIÓN DE REQUISITOS	149
5.7.1	Verificación y análisis de consistencia entre modelos.	149
5.7.1.1	Matriz de Requisitos de usuario de capacidad-Casos de uso.....	149
5.7.1.2	Matrices de Requisitos software-Casos de uso.....	150
5.7.2	Validación de los modelos	152
6.	DISEÑO DEL SISTEMA	152
6.1	INTRODUCCIÓN.....	153
6.1.1	Propósito	153
6.2	ALCANCE	153
6.3	DEFINICIÓN DE LA ARQUITECTURA	155
6.3.1	Definición de los niveles de la arquitectura	155
6.3.2	Especificación de estándares y normas de diseño y construcción.....	155
6.3.2.1	Nombres de Ficheros.....	155
6.3.2.2	Idioma.....	155
6.3.2.3	Cabeceras de las clases	156
6.3.2.4	Visibilidad de las clases.....	156
6.3.2.5	Identificadores de las clases.....	156
6.3.2.6	Constantes	156
6.3.2.7	Variables	156
6.3.2.8	Funciones y procedimientos	156
6.3.3	Especificación del entorno tecnológico	157
6.3.3.1	Hardware	157
6.3.3.2	Software.....	157
6.3.3.3	Comunicaciones.....	157
6.3.4	Especificación de requisitos de seguridad y operación	157
6.3.5	Estudio de la seguridad requerida en el proceso de diseño del sistema.....	157
6.4	DISEÑO DE LA ARQUITECTURA DE SOPORTE.....	158
6.5	DISEÑO DE CASOS DE USO REALES	160
6.5.1	Identificación de clases asociadas a un caso de uso	160
6.5.2	Diseño de la realización de los casos de uso	161
6.6	DISEÑO DE CLASES.....	162
6.6.1	Identificación de clases de diseño	163
6.7	DISEÑO FÍSICO DE LOS DATOS.....	169
6.8	VERIFICACIÓN Y ACEPTACIÓN DE LA ARQUITECTURA DEL SISTEMA.....	170
6.8.1	Análisis de consistencia de las especificaciones de diseño	170
6.9	ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS.....	171
6.10.1	Especificación de Requisitos de Implantación.....	172
7.	PLAN DE VALIDACIÓN Y VERIFICACIÓN DEL SOFTWARE	173
7.1	INTRODUCCIÓN.....	173



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

7.1.1	Propósito del documento.....	173
7.1.2	Visión general del plan de Validación y Verificación del Sistema	173
7.2	PLAN DE PRUEBA.....	175
7.2.1	Elementos de prueba.....	175
7.2.2	Características que se probarán	176
7.2.3	Entregables	177
7.2.4	Tareas de prueba.....	177
7.2.5	Necesidades del entorno	178
7.2.6	Criterio de aceptación/rechazo de cada caso de prueba.....	178
7.3	PROCEDIMIENTOS DE PRUEBA	179
7.4	PLANTILLA DE INFORME DE PRUEBAS.....	188
7.5	MATRIZ DE TRAZABILIDAD PRUEBAS/REQUISITOS DE CAPACIDAD	189
8.	IMPLEMENTACIÓN DEL SISTEMA	191
8.1	INTERFAZ APLICACIÓN.....	191
8.2	INTERFAZ USUARIO	192
8.3	INTERFAZ COM.....	193
8.4	INSTRUCCIONES.....	194
8.5	IA DE COM.....	195
8.5.1	Responder.....	195
8.5.2	Preguntar	195
8.5.3	Descartar	196
8.5.4	Resolver	196
9.	EXPERIMENTACIÓN Y RESULTADOS	197
9.1	VERIFICACIÓN DE REQUISITOS. INFORME DE PRUEBAS.....	197
9.1.1	Informe de pruebas.	197
9.2	TEST DE VALORACIÓN	221
9.2.1	Conclusiones test de valoración	225
10.	CONCLUSIONES	227
11.	LÍNEAS FUTURAS.....	228
LISTA DE ACRÓNIMOS.....		229
BIBLIOGRAFÍA		230
ANEXO A: MANUAL DE LA APLICACIÓN PFC - ¿QUIÉN ES?		232
1	INICIAR JUEGO.....	232
2	DESARROLLO DE LA PARTIDA.....	233



1. INTRODUCCIÓN

El presente documento técnico describe el proceso de desarrollo de un software, específicamente, una aplicación integrada en la red social *Facebook*. Primero se realizará una descripción del contexto actual, así como, se detallarán los objetivos a alcanzar durante dicho desarrollo y, finalmente, se describirán los distintos apartados de los que consta el documento.

1.1 Contexto

El éxito producido hace pocos años por la aparición de las redes sociales no sólo ha sido originado por satisfacer la necesidad de las personas de relacionarse entre sí, sino que el éxito también proviene de la manera en la que la comunicación es realizada. Las redes sociales ponen a disposición de los usuarios una serie de herramientas (chats, mensajería, grupos, eventos, compartir fotos, etc.) que facilitan la comunicación e interacción de unos usuarios con otros. Pero sin duda la red social reina en la categoría de herramientas puestas a disposición de sus usuarios es *Facebook*. Ésta tiene una plataforma que facilita a los programadores la creación de aplicaciones capaces de ser integradas en dicha red social facilitando la comunicación entre su base de datos y la RIA (*Rich Internet Applications*) creada a través de una API (*Application Programming Interface*). De modo que las posibilidades para esta red social se disparan, teniendo a una comunidad entera de empresas y programadores creando RIAs de diversa índole, haciendo que prácticamente sean infinitas las formas de comunicación o de interactuar entre los usuarios.

Este hecho motivó el interés de crear una RIA en *Facebook* y aprovechar las herramientas que ésta pone a disposición de los programadores para que la experiencia del usuario final sea la más satisfactoria posible.

Después de valorar varias tecnologías para la realización de la RIA, fue seleccionada *Flex* de *Adobe*, la cual aúna *ActionScript* y *MXML* dando muchas posibilidades al programador a la hora de realizar aplicaciones. La herramienta más famosa que soporta dicha tecnología, y la cuál ha sido usada para la implementación de la aplicación, es *Flash Builder* en su versión 4.0, también perteneciente a *Adobe*, la cual posee un gran potencial para la creación de este tipo de software.

Por último, a la hora de escoger que tipo de aplicación se quería implementar, había que contar con la cantidad ingente de aplicaciones ya creadas para *Facebook*, por lo que no era tarea fácil. Finalmente se decidió realizar una nueva versión del juego de mesa “¿Quién es quién?” que aprovechara la posible comunicación con las bases de datos de *Facebook* para así obtener información de los usuarios presentes en la aplicación, de manera que la experiencia con la RIA sea completa.



1.2 Objetivos

Los objetivos que se pretenden alcanzar mediante la realización de este proyecto se detallan a continuación:

- El objetivo principal reside en la creación de una aplicación basada en el juego de mesa “¿Quién es quién?” que sea capaz de cubrir todas las necesidades que plantea el cliente.
- Se busca que el software interactúe con la red social *Facebook*, mediante la realización de consultas a las tablas alojadas en la base de datos de dicha red social. Dichos datos sería utilizados como un elemento más del juego, ya que estas consultas estarían basadas en los datos personales, o de perfil, del usuario/jugador y de los amigos de éste, pudiendo realizar cuestiones sobre estos datos.
- La partida se realizará contra la máquina por lo que sería conveniente diseñar una inteligencia artificial para que la máquina pudiera presentar resistencia a la hora de jugar y así hacer más amenas las partidas.
- También se persigue la creación de un entorno intuitivo el cuál facilite el juego a los usuarios finales de la aplicación para así obtener una partida fluida. Además, se le proporcionará un manual con las instrucciones del juego, así como el manejo de la interface.
- Para la realización del proyecto se seguirá la metodología de desarrollo *Métrica Versión 3* con el fin de generar software de calidad.
- Respetar la planificación del proyecto establecida al principio del proyecto, salvo ligeras modificaciones, que permitan cumplir los plazos de entrega.
- Poner en práctica los conocimientos adquiridos durante la realización de la carrera, ampliando los conocimientos sobre lenguajes informáticos y la programación orientada a objetos.
- Proporcionar información necesaria para el mantenimiento y ampliación de la aplicación.

1.3 Estructura del Documento

El presente documento recopila toda la información empleada durante el desarrollo del proyecto, de manera que se sigue la siguiente estructura:

- Introducción: Explica cómo surge el proyecto, cuáles son los objetivos a alcanzar con su realización, y la estructura de la documentación presentada.



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

- Estado del arte: describe la investigación llevada a cabo con el fin de conocer y diferenciar las RIAs existentes en la actualidad, así como las distintas redes sociales.
- Estudio de la Viabilidad del Sistema: Realiza un estudio de la solicitud del cliente y de la situación actual proponiendo una solución inicial al problema planteado por el cliente en base a las restricciones económicas, técnicas, legales y operativas.
- Gestión del Proyecto: Detalla todos los aspectos relacionados con la gestión del proyecto, tales como los recursos necesarios, una planificación detallada, y una estimación de los costes que supone su realización.
- Análisis del Sistema: Recoge el conjunto de requisitos que ha de cumplir el sistema a construir.
- Diseño del Sistema: Define el diseño del sistema de forma exhaustiva y con un nivel de detalle profundo. También realiza un estudio de toda la tecnología útil para la realización del sistema.
- Plan de Validación y Verificación del Software: Ofrece un catálogo de pruebas de aceptación que servirán para comprobar el correcto funcionamiento del sistema y para verificar el cumplimiento de las funcionalidades requeridas por el cliente.
- Implementación del Sistema: Detalla la fase de codificación de las distintas interfaces del sistema desarrollado, así como de la IA a la que hará frente el usuario.
- Experimentación y resultados: Presenta la realización de pruebas para la verificación de requisitos, además de la realización de un test de valoración a una serie de usuarios.
- Conclusiones: Resume las conclusiones alcanzadas tras la realización del presente proyecto.
- Líneas futuras: Presenta posibles mejoras del software de cara a un futuro.
- Anexos: Incluye un documento con las instrucciones del juego para poder desarrollar una partida con total fluidez.



2. ESTADO DEL ARTE

En este apartado se describe la investigación llevada a cabo con el fin de conocer y diferenciar las RIAs existentes en la actualidad, las herramientas para el desarrollo de estas, así como las distintas redes sociales presentes en internet. Finalmente se realizará una aplicación sencilla para probar el funcionamiento de las distintas herramientas.

2.1 Introducción

Las redes sociales son el último fenómeno de internet, con más de 800 millones de usuarios entre todas las redes existentes, permitiendo a estos millones de personas establecer contacto con personas con las que lo habían perdido, crear nuevas amistades o, simplemente, compartir fotos o vivencias con el resto de usuarios (Redacción Puro Marketing, 2008).

Pero ha habido redes sociales que han querido dar un paso más allá, permitiendo la creación de software por parte de los propios usuarios o terceros que funcione bajo dichas redes. Esto lo convierte en todo un éxito que ha abierto nuevas puertas tanto de negocios como de tecnologías, permitiendo a empresas desarrollar productos enfocados a estas redes. El mejor ejemplo de ello es *Facebook*, que con más de 600 millones de usuarios se ha convertido en la red social por antonomasia, creando un flujo de negocio centrado en la creación y explotación de software que funciona utilizando recursos que el propio *Facebook* pone a disposición de los desarrolladores, haciendo que la experiencia de integración del programa con la red social sea amplia.

¿Pero porqué del éxito de ésta frente a sus competidoras? La clave reside en la flexibilidad que ofrece esta red para realizar dicho software por terceros y poder implementar éstos fácilmente en *Facebook*, lo que produce que la experiencia del usuario con esta red sea casi infinita. Juegos, utilidades, visores de noticias... son miles los programas creados para que funcionen con *Facebook* y son miles los usuarios que hacen gala de ello.

En este documento se recogerá el desarrollo histórico de las redes sociales y en especial de la red más popular, *Facebook*. Posteriormente, se explicarán los pasos esenciales para la realización de software que funcione dentro de la red social y como alojarlo en un servidor gratuito de internet, mostrando las herramientas que ofrece *Facebook* para el desarrollo de éstos. Finalmente, se dará a conocer la última versión de la tecnología presentada por *Adobe*, el *Flash Builder 4*, y se realizará un ejemplo que explote gran parte de las cualidades de dicha tecnología integrándose en *Facebook*.



2.2 Introducción a las redes sociales

Internet ha sido y es un punto para la interactividad social. En un inicio, las redes de ordenadores eran vistas por el público general como algo extraño y casi militar. Pero a medida que se expandió más allá de unos pocos núcleos privilegiados, también lo hizo la idea de que los ordenadores interconectados podrían servir de gran foro para discutir temas de interés mutuo, y quizás incluso para conocer o reencontrarse con otras personas. En la década de 1970, ese proceso comenzó en serio (Van Camp, 2011).

A finales de los 70 y principios de los 80, los ordenadores eran objetos muy raros. El lenguaje de las máquinas era desconcertante, y su potencial aparentemente limitado. La sociedad veía como un acto de aislamiento o antisocial el sentarse delante de un ordenador. Sin embargo, estos “antisociales”, fueron los que ayudaron a que las redes sociales comenzaran a dar sus primeros pasos para convertirse en el fenómeno cultural que conocemos hoy.

Todo comenzó con *BBS (Bulletin Board System)*, unos lugares de encuentro en línea que fueron creados a través de trozos de código de forma independiente que permitían a los usuarios comunicarse con un sistema central donde se podían descargar archivos o juegos (muchas veces incluidos los programas informáticos piratas) y enviar mensajes a otros usuarios. Se accedía a través de un modem conectado a la línea telefónica. *BBS* era dirigido por aficionados que mimaban los aspectos sociales y de interés específico de sus proyectos. Por otra parte, se estimularon las reuniones de personas entre los usuarios de *BBS*, denominadas popularmente como “quedadas”, por lo que de pronto lo que antes era antisocial se había convertido en social.

Aunque la tecnología de la época restringía la flexibilidad de estos sistemas y la experiencia del usuario final, además de una velocidad de transferencia mínima, *BBS* continuó ganando popularidad en los años 80 hasta bien entrados los años 90, cuando Internet comenzó realmente a funcionar.

Pero había otras opciones para la interacción social mucho antes de que se produjera la explosión de internet a nivel global. Una de ellas fue *CompuServe*, un servicio que comenzó en la década de los 70 como una empresa orientada a ofrecer soluciones de servidores para redes de ordenadores, aunque su expansión no llegaría hasta la década de los 80.

CompuServe permitía a sus miembros compartir archivos y acceder a noticias y eventos. Además, permitía el envío de emails, esto ayudó a popularizar el uso del email, ya que durante aquella época esta tecnología era muy poco utilizada por el público en general. Los usuarios también podían unirse a cientos de foros de discusión donde poder exponer su opinión acerca de un tema en concreto, al igual, que leer la de otros miembros de la comunidad. Estos foros tuvieron un éxito enorme.

AOL (América OnLine) es considerada como una de las precursoras en el mundo de las redes sociales. *AOL* permitía búsquedas a través de los “Perfiles de los Miembros”, en las cuales los usuarios obtendrían una lista con los detalles pertenecientes a los usuarios, convirtiéndose así en el servicio más relevante.



Sin embargo, Internet no paraba de evolucionar... *Yahoo* abría su tienda, *Amazon* comenzaba la venta de libros, y empezaba a popularizarse el PC llegando a todos los hogares.

En ese momento surgió *Classmates.com* con un éxito casi inmediato. Esta red ayudaba a que un usuario pudiese localizar a sus antiguos compañeros de colegio o instituto. Aún hoy en día el servicio cuenta con unos 40 millones de cuentas registradas.

SixDegrees.com no alcanzó el mismo nivel de éxito. Su nombre estaba basado en la teoría de que de alguna manera todos estamos conectados con el actor Kevin Bacon por seis grados (pasos) o menos. El sitio surgió en 1997, y fue uno de los primeros en permitir a sus usuarios crear perfiles, invitar a amigos, organizar grupos y perfiles de navegación de otros usuarios. Poco a poco el sitio se transformó en una asociación informal de usuarios y recibieron numerosas quejas de spam. *SixDegrees.com* terminó su andadura justo después del año 2000.

Un producto de *Community Connect Inc.*, fundada un año antes en el apartamento de Nueva York del ex banquero de inversiones, y futuro CEO (del inglés *Chief Executive Officer*, director ejecutivo) de *Community Connect*, Ben Sun, fue *AsianAvenue.com* seguido por *BlackPlanet.com* en 1999, y por *MiGente.com*, orientada a los hispanos, en 2000. Redes sociales enfocadas a unos grupos sociales muy concretos. Los tres han sobrevivido hasta el día de hoy. *BlackPlanet.com*, en particular, goza de gran éxito a lo largo de todo este tiempo. De hecho, actualmente atrae a más de tres millones de visitantes cada mes.

Desde ese momento se han creado diversas redes, de las cuales unas permanecen y otras han desaparecido. Según la zona geográfica, el líder puede ir cambiando, pero hasta el 2009 los principales competidores a nivel mundial eran: *Hi5*, *MySpace*, *Facebook*, *Tuenti*, *Twitter* y *Orkut*.

MySpace se instaló en 2003, y se ha diferenciado de otros sitios en que permite a los usuarios personalizar sus páginas. Los adolescentes fueron los primeros en adoptarlo y creció gracias a estos. *MySpace* incluso ha modificado su política de usuario para permitir a los menores de edad crear sus propios perfiles a raíz de la demanda de este sector. Conforme el sitio ha crecido, tres poblaciones se empezaron a formar: los músicos/artistas, jóvenes y adultos en su etapa posterior a la universidad.

Facebook fue creado originalmente para apoyar a las redes universitarias. En 2004 los usuarios del sitio estaban obligados a proporcionar las direcciones de correo electrónico asociada con las instituciones educativas. Este requisito lo llevó a tener un modelo de acceso relativamente cerrado, por lo que los usuarios tenían la percepción de la web como un sitio íntimo y privado de una comunidad cerrada. *Facebook* posteriormente se ha ampliado para incluir a los estudiantes de secundaria, profesionales, y finalmente todos los usuarios potenciales de Internet. Otra característica que distingue a *Facebook* es la capacidad para desarrolladores externos de crear aplicaciones, lo que permite a los usuarios personalizar sus perfiles y realizar otras tareas. Esto abre un sinfín de posibilidades para esta red social, y probablemente sea este hecho el que haya convertido a *Facebook* en la número uno de todas las redes sociales. Más adelante se detallará como crear aplicaciones para *Facebook* y las herramientas que ofrecen para realizar éstas.



Cronología:

Varios han sido los fines que han movido la creación de redes sociales a través del tiempo. Desde 1997 hasta hoy el más representativo ha sido diseñar un lugar para el encuentro de miles de personas que tengan intereses en común (Van Camp, 2011).

1. En 1997 se creó *Sixdegrees*. Este fue el primer servicio que permitió crear un perfil en la web, agrupar a los contactos directos e intercambiar mensajes con ellos.
2. En 1999 se creó *LiveJournal.com*. Este fue uno de los primeros servicios de redes sociales en ofrecer blogs y diarios en línea.
3. En el 2001 aparece *Ryze.com* y nace con ella la primera red social especializada. En la actualidad ésta agrupa por lo menos a 500 mil profesionales.
4. En el 2002 nace *Friendster*, que fue la primera en tener un sistema inteligente capaz de relacionar a los usuarios de la red según sus gustos.
5. También en este año nace *Fotolog*, la primera red social que nace con el fin de intercambiar fotografías.
6. *Last FM* nace en el 2002 con el objetivo inicial de ser una emisora en línea, pero con el tiempo se convierte en una red social movida por el interés en la música.
7. En el 2003 se crea *Linked in*, una red social con el fin de propiciar las relaciones empresariales.
8. En el 2003 nace *My Space*, tal vez la red más grande del momento agrupando a más de 250 millones de usuarios.
9. En ese mismo año nace *MSN Spaces* llamado hoy *Windows Live Spaces*, que tiene 100 millones de visitantes al mes.
10. También en el 2003 nace *Hi5* que, hoy por hoy, tiene 50 millones de cuentas activas.
11. *Flickr* nace en el 2004 como una red social que funciona a través del servicio de compartir fotografías.
12. En este año también nace *Orkut*, la red social de *Google* con gran popularidad en Brasil y La India.
13. *Facebook*, la sensación del momento, nace en el 2004, y tiene más de 600 millones de cuentas activas. Cuenta con la ventaja de que desarrolladores externos pueden crear aplicaciones para la red social, lo que permite a los usuarios personalizar sus perfiles, jugar, comunicarse... Este es el motivo más probable de que *Facebook* sea la número uno de las redes sociales. En este documento detallaremos como crear aplicaciones para *Facebook*.



14. En el 2005 nace *Yahoo 360*, la red social de esta empresa que cuenta con servicio de correo, calendario, fotos, eventos, blog y música.

15. En el 2005 también nace *Bebo*, una red muy básica con el fin de contactar amigos e intercambiar archivos multimedia.

16. Y la última en la cadena ha sido *Xing*, esta nació en el 2006 con el fin de agrupar ejecutivos permitiéndoles compartir datos, crear agendas conjuntas y construir discusiones en torno a un tema común.

En la figura 1 se puede observar el uso de las redes sociales más importantes.

Nombre	Descripción	Usuarios registrados
Badoo	Popular en Europa	37,000,000
Classmates	Colegio, universidades, trabajo y militares	50,000,000
BlackPlanet	Afroamericanos	20,000,000
MySpace	Sitio HTML	130,000,000
Facebook	Popular a nivel mundial	600,000,000
Flickr	Fotolog	32,000,000
Tuenti	Muy popular en España	4,500,000
Twitter	Micro-blog, RSS	75,000,000
WindowsLive Space	Blogging	120,000,000
Orkut	Propiedad de Google	180,000,000
hi5	Popular en la India, Portugal, Rumanía África central y América latino.	80,000,000
NetLog	Popular en Europa, el mundo árabe y la provincia del Québec canadiense	59,000,000

Figura 1. Ejemplos de redes sociales, una pequeña descripción y número de usuarios.



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

Como se puede observar es *Facebook* la red social que cuenta con más usuarios en todo el mundo. Básicamente, se achaca esta superioridad a la posibilidad de desarrollar aplicaciones que funcionan bajo *Facebook*, ampliando la experiencia que esta red social puede ofrecer en comparación con otras.



2.3 Facebook

Facebook es un sitio web gratuito de redes sociales creado por Mark Zuckerberg. Originalmente era un sitio para estudiantes de la Universidad de Harvard, pero actualmente está abierto a cualquier persona que tenga una cuenta de correo electrónico. Los usuarios pueden participar en una o más redes sociales, en relación con su situación académica, su lugar de trabajo o región geográfica.

Ha recibido mucha atención en la blogosfera y en los medios de comunicación al convertirse en una plataforma sobre la que terceros pueden desarrollar aplicaciones y hacer negocio a partir de la red social. A pesar de ello, existe la preocupación acerca de su posible modelo de negocio, dado que los resultados en publicidad se han revelado como muy pobres.

Esto ha provocado la irrupción de un nuevo perfil laboral, *community manager*, un profesional que aúna las labores propias de un relaciones públicas con las de un generador de contenidos a través de internet.

A mediados de 2007, *Facebook*, lanzó las versiones en francés, alemán y español para impulsar su expansión fuera de Estados Unidos, ya que sus usuarios se concentraban en Estados Unidos, Canadá y Gran Bretaña. En enero de 2010, *Facebook* contaba con 350 millones de miembros, y traducciones a 70 idiomas.

2.4 Facebook Developers

Facebook logró su popularidad siendo una red social capaz de contener aplicaciones realizadas por terceros, permitiendo así la realización de distintas actividades a partir de la misma. Más allá de las aplicaciones propias que presenta, como es el caso de *The Wall*, página a modo de pizarra virtual del usuario donde otros pueden dejarle mensajes, o la aplicación *Status*, que permite a los usuarios indicar sus actividades o estados de ánimo al resto de la red, *Facebook* permite la creación de aplicaciones mediante la utilización de *web services* para el acceso a los datos de la red.

En el año 2006 fue presentada la *Facebook Platform*, para que sus propios desarrolladores pudieran crear aplicaciones que permitieran a los usuarios compartir y permanecer conectados a la red social. Hoy en día, cerca de un millón de desarrolladores están ayudando a hacer la Web aún más social.

Facebook Developers es una plataforma de apoyo a todos los usuarios que decidan crear una aplicación para *Facebook*, ofreciendo todo lo necesario para poder compartir nuestra aplicación. Esta plataforma cuenta con *Wiki Developers Facebook*, una guía con estructura de *Wikipedia* donde se puede encontrar información en abundancia, desde cómo acceder a los datos de los usuarios que *Facebook* ofrece para poder ser usados por las aplicaciones, hasta los pasos pertinentes para poder compartirla.

Más de 600 millones de usuarios de *Facebook* tienen a su alcance unas 20.000 aplicaciones realizadas para esta red social. Aquí se recogen algunas aplicaciones creadas por desarrolladores externos o usuarios de *Facebook* (PC World digital, 2009).

Farmville

Este juego ha sobrepasado los 83 millones de usuarios. Se trata de un videojuego cuya finalidad consiste en crear una empresa (granja) y hacer que prospere mediante una interfaz agradable, como se puede observar en la figura 2. Este juego aprovecha la posible interacción entre los usuarios de la red social a la perfección. Por ejemplo, si se tienen problemas, se puede pedir ayuda a los vecinos, o sea, a los contactos que un usuario tenga en la red social que también se hayan unido al juego. Aquí radica el éxito de *Farmville*, pues estos juegos prevén que los participantes se encarguen de cuidar la tierra de los amigos que están off-line. Por lo tanto, cuantos más amigos se tengan, mejor clasificación se obtendrá.



Figura 2. Imagen del juego *Farmville*.

Dropio.io

Dropio.io ofrece 100 MB de espacio gratuito y anónimo para compartir ficheros con amigos de *Facebook* de manera sencilla como se puede observar en la figura 3. Espacio gestionado por el usuario, incluyendo los permisos que tendrán sobre los archivos o si un “visitante” del perfil puede borrar o añadir archivos. Cuenta con 1.674 usuarios activos al mes.

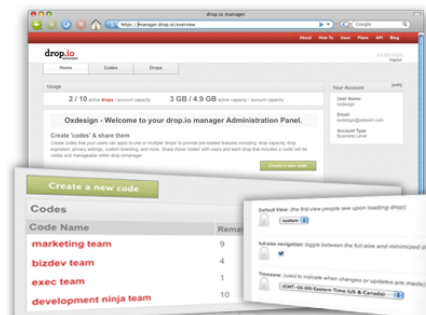


Figura 3. Capturas de la aplicación *Dropio.io*.

Mob Wars

Mob Wars permite al usuario llegar a convertirse en el nuevo *Don Vito* de *Facebook*, figura 4. Sólo hay que armarse comprando armas en la tienda del juego, hacer los “trabajos” para conseguir dinero y experiencia, invertirlo en propiedades y luchar contra bandas rivales representadas por otros usuarios de la red social. Tiene unos 2.315.118 usuarios activos al mes y está traducido al castellano.



Figura 4. Juego de mafia *Mob Wars*.

KaChing

KaChing presta 10.000 \$ virtuales para crear una estrategia de inversión y gestionar una cartera de acciones comprando y vendiendo con dinero virtual a precio de mercado llegando a enriquecerse con los beneficios, figura 5. Esta aplicación tiene 36.514 usuarios activos por mes.



Figura 5. *KaChing*.

YouTube Video Box

Aunque *Facebook* dispone de su propia aplicación de vídeo, y a pesar de que recientemente ha incorporado calidad HD, *YouTube* sigue siendo el rey indiscutible en ese terreno, figura 6. Con esta aplicación se podrá añadir videos de *YouTube* al perfil, comentarlos, darles una puntuación, compartirlos con los amigos y ver qué videos son sus favoritos. Cuenta con 1.402.299 usuarios activos/mes. También se puede encontrar en castellano.



Figura 6. Aplicación de *YouTube* para *Facebook*.

Cities I've visited (Ciudades que he visitado)

Imprescindible para viajeros, con esta aplicación se puede señalar en un mapa *Google* las ciudades en las que uno ha estado, aquellas que visitará próximamente y sus favoritas, como se observa en la figura 7. Esta información se puede compartir con los amigos y aprovechar las experiencias de otros viajeros, por ejemplo, en cuanto a su opinión sobre determinado hotel o recomendaciones de todo tipo. Son 1.439.955 los usuarios activos al mes de esta aplicación que puede usarse en castellano. (Hodge, 2008)



Figura 7. Mapa con las ciudades que se han visitado.



2.5 Perfil del usuario de los social games

“La posibilidad de colaborar con los vecinos es, precisamente, lo que convierte a los *social games* en más avanzados y seductores que los videojuegos habituales”, explica Giuseppe Riva, profesor de Psicología y Nuevas Tecnologías de la Comunicación en la Universidad Católica de Milán. “Cuando jugamos, de hecho, intentamos satisfacer nuestras necesidades. Los juegos sociales, además de saciar el impulso de sentirnos realizados mediante la exhibición de nuestras habilidades – típico en todos los videojuegos –, nos permiten cubrir otras dos más: la autoestima, porque el resultado es visible también para nuestras amistades, con lo que podemos presumir ante ellos de la puntuación, y la necesidad de asociación, ya que el hecho de poder ayudar a un amigo a desenvolver sus tareas en el juego refuerza las relaciones sociales, una necesidad atávica en el hombre”. (Marketing Capacitación, 2011)

Según un estudio realizado por la empresa *Popcap Game* en Estados Unidos y Gran Bretaña, el 55% de los jugadores de *social games* son mujeres. La edad media es de 43 años. En EE.UU., el 46% de los jugadores tiene más de 50 años y sólo el 6% es menor de 21 años. Pero hay más: el 95% de los entrevistados juega varias veces por semana y en el 59% de los casos una partida dura 30 minutos –sólo el 9% resiste más de 3 horas–. Más del 56% lleva jugando más de un año y el 83% lo hace a través de *Facebook*. Se juega por diversión (53%), para liberarse del estrés (45%), por competitividad (43%) y para evadir la mente (32%). Pero el 24% usa los juegos sociales para estar en contacto con amigos. De hecho, la amistad es la base de los social games. El 62% de las personas prefiere interactuar con sus propios amigos y el 57% de los jugadores ha sabido de su existencia al recibir la invitación de alguno de ellos. Y no se abandona: el 39% practica entre 3 y 5 títulos, y en el 28% de los casos se gasta dinero para enriquecer las propias opciones de juego. (Marketing Capacitación, 2011)

En este sentido, el profesor Riva advierte del lado oscuro de los *social games*: “Son muy atractivos, lo que hace que personas con baja autoestima se refugien en ellos, una conducta que dificulta su contacto con el exterior. Por otra parte, al ser una gran fuente de distracción, los juegos sociales pueden interferir en el rendimiento laboral”. Es por eso por lo que muchas empresas tienen bloqueado el acceso de sus trabajadores a las redes sociales. De hecho, la revista *Time* publica que *Farmville* está considerado como el noveno peor invento de la historia. (Marketing Capacitación, 2011).

Ahora se pasará a detallar como realizar una aplicación sencilla para *Facebook*.



2.6 Desarrollo de Aplicaciones con Facebook Developers

Básicamente *Facebook developers* es una API (del inglés *application programming interface*), interfaz de programación de aplicaciones, que permite la creación de aplicaciones mediante la utilización de *web services* para el acceso a los datos de la red, así, los desarrolladores pueden explotar el potencial de la red social.

Una API es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Además, por ser una interfaz de programación representa una interfaz de comunicación entre componentes de software (*Facebook* y nuestro software). De manera que se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general. De esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio.

2.6.1 Creando aplicaciones en Facebook

Los recursos disponibles a los desarrolladores se agrupan en tres categorías (Facebook Developers, 2007):

- **API:** es una interfaz basada en *REST* (Representational State Transfer - técnica de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web) que permite el acceso a los datos del perfil, amigos, fotos y eventos del usuario mediante la utilización de mensajes *GET* o *POST*.
- **Consultas (*FQL*, “*Facebook Query Language*”):** es un lenguaje de consultas similar a *SQL* utilizado para acceder a los mismos datos que la API pero permitiendo consultas más complejas.
- **Maquetación (*FBML*, “*Facebook Markup Language*”):** es un lenguaje similar a *HTML* que permite ser intercalado con éste para integrar las aplicaciones en *Facebook*. Utilizando *FBML* se puede acceder, por ejemplo, al perfil, acciones del perfil y *feeds*. Este lenguaje también incluye soporte para *AJAX* y *Javascript*.

2.6.2 Estructura de una aplicación en Facebook

Una aplicación en *Facebook* se compone:

- **Product Directory:** cuando un usuario navega por el directorio de aplicaciones de *Facebook*, por cada aplicación se muestra una pequeña sección con el nombre, una imagen y una pequeña descripción.
- **About:** esta página muestra información general de la aplicación. Debe inducir al usuario a instalar la aplicación.



- **Left Nav:** es el panel de navegación izquierdo. Las aplicaciones pueden tener su icono y nombre en este panel.
- **Canvas Page (Lienzo):** es la página principal de la aplicación.
- **Home:** esta página es accedida utilizando el panel de navegación de la izquierda, en general muestra información de los amigos del usuario.
- **User Dashboard:** es la página donde el usuario administra sus datos en una aplicación y determina de qué forma las aplicaciones acceden a sus datos personales.
- **Profile:** es la representación online de la identidad del usuario. La API provee múltiples puntos de integración con el perfil del usuario permitiendo actualizar datos del mismo.
- **Profile Box:** presenta información actualizada sobre las acciones recientes del usuario.
- **Profile Actions Links:** debajo de la foto del usuario en el perfil, se pueden agregar enlaces para invocar acciones en la aplicación.
- **Privacy Settings:** se utiliza para definir los niveles de privacidad de los datos del perfil por aplicación.
- **News Feed:** como su nombre indica es un *feed* de noticias, allí las aplicaciones pueden “publicar” información del usuario de la misma.
- **Alerts:** las aplicaciones pueden enviar notificaciones a los usuarios a través del correo electrónico.
- **Message Attachments:** las aplicaciones pueden incluir archivos adjuntos que aparecen en la ventana para componer mensajes.
- **Requests:** las aplicaciones pueden crear peticiones que aparecen en la parte superior izquierda de la página principal. En general son iniciados por amigos que solicitan realizar alguna acción.

2.7 Introducción a Developers. Prueba.

Con el fin de explicar los pasos a seguir para crear una aplicación, en este punto se creará una aplicación de ejemplo. La aplicación que se va a realizar es muy sencilla, ya que lo que se quiere mostrar en esta parte del documento son los pasos básicos para poder colgar una aplicación en *Facebook*, básicamente esta aplicación mostrará el mensaje por pantalla: “Prueba superada”. Para ello se debe acudir a la web de *Developers*: <http://developers.facebook.com>. Como se aprecia en la figura 8, después de darse de alta en su sistema, hay que entrar en la página de *Desarrollador* personal.

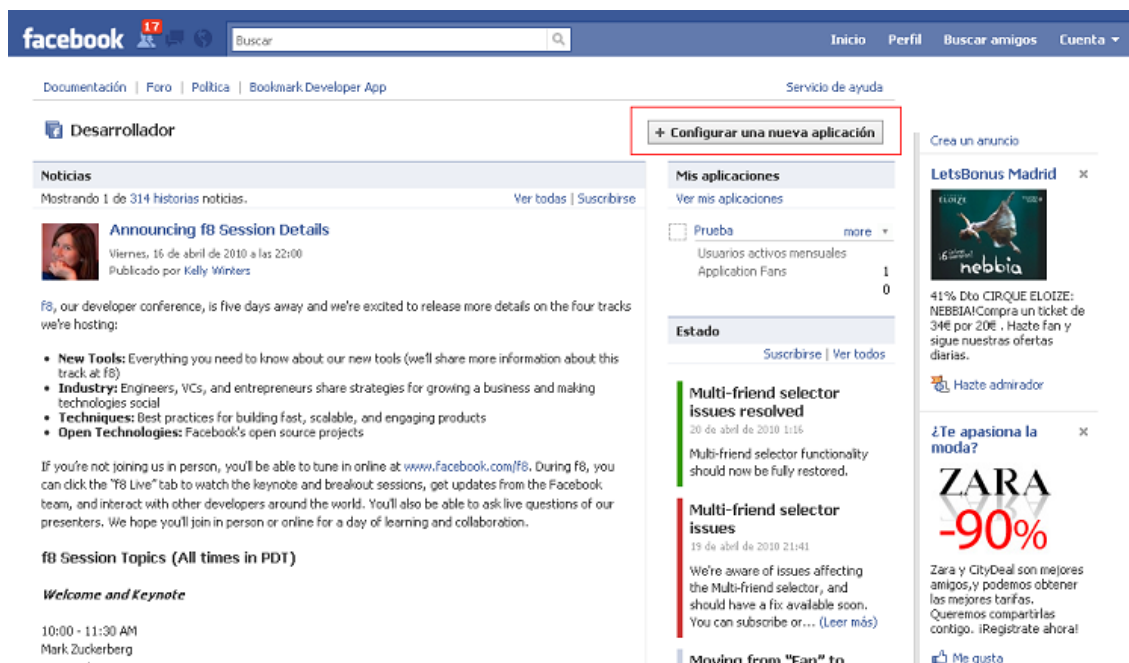


Figura 8. Página principal de usuario de *Facebook Developer*.

A continuación se ha de presionar el botón *Configurar una nueva aplicación*: allí se preguntará el nombre de la aplicación y el usuario ha de aceptar las condiciones de *Facebook*, figura 9.



Figura 9. Aceptar condiciones de *Facebook*.

Una vez aceptadas las condiciones, aparecerá la página que se observa en la figura 10, donde se muestra el nombre de la aplicación escrito anteriormente, y se facilitará la clave API y el *Secreto*, esos dos datos son los que permiten conectar la aplicación con la interfaz de *Facebook*, así que son sumamente importantes. Pero en esta prueba no se hará uso de estas. También se podrá seleccionar un logotipo, icono y descripción, pero serán campos u opciones no obligatorias.

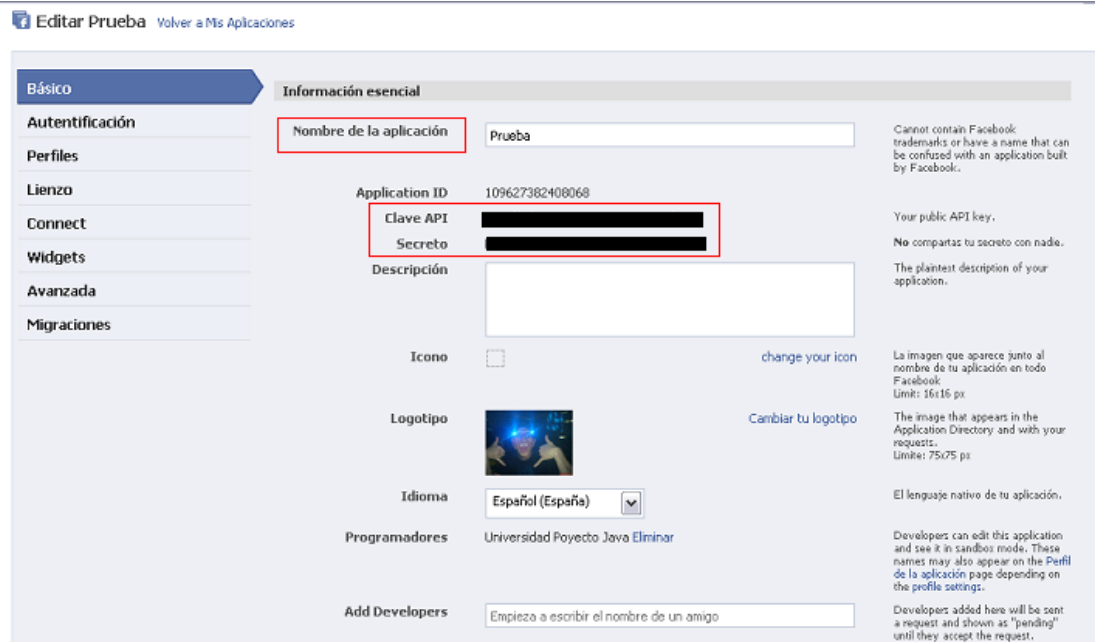


Figura 10. Página principal de configuración de la aplicación.

El usuario ya está dado de alta en *Developers*, ahora se necesita un servidor donde poder alojar la aplicación. Una solución rápida, gratuita y sencilla, es hacer uso de los server gratuitos que se pueden encontrar, figura 11. Para este documento se ha usado el host de <http://www.miarroba.com>.



Figura 11. *Miarroba.com* ofrece hosting gratuito.



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

Una vez dado de alta el nuevo usuario, y después de haber seleccionado el nombre de la web, se mostrará información relevante tal como viene reflejado en la figura 12.

The screenshot displays a hosting control panel with three main sections:

- INFORMACIÓN GENERAL:** Contains fields for URL (pruebaproject.webcindario.com), Title (project), Description (Pagina de prueba), Date of creation (16 de Abril de 2010), PHP5 (No), MySQL (No), Home (/home/webcindario/pr/r/pruebaproject), Error 404 (No especificado), and Page index (No).
- ESPACIO OCUPADO EN DISCO:** Shows disk usage statistics: Actualizado (16/Abr/2010 11:18 GMT+1), Ficheros (0 B), MySQL (0 B), Total (0 B), and Disponible (100.00 MB). A green progress bar indicates 0.00% usage.
- DATOS DE ACCESO CLIENTE FTP:** Provides FTP access details: Dirección (ftp.webcindario.com), Usuario (pruebaproject), and Contraseña (*****).

A sidebar on the right contains various links categorized under Encuestas, Servicios, and General.

Figura 12. Información general sobre el hosting.

Se podrán ver todos los datos relacionados con el server. Después se debe subir el *index.html*, que será la página de inicio de la aplicación, figura 13.

The screenshot shows a file manager interface with a top navigation bar, a toolbar, a file list, and a sidebar.

- Top Navigation:** Includes a 'Volver' button and a link to 'Volver al panel de control'.
- Toolbar:** Contains icons for Information, Preferences, PHP, MySQL, WebFTP, Estadísticas, Eliminar, and Ayuda.
- File List:** Displays a table with columns for selection, file name, size, and date. The file 'index.html' (33 B, 16/04/2010 11:38) is listed.
- Bottom Bar:** Includes buttons for 'Nuevo', 'borrar', 'Renombrar', 'descargar', and 'nueva carpeta'.
- Sidebar:** Contains a list of links categorized under Webmasters, Espacio Web, and Servicios.

Figura 13. Gestor de archivos de *Miarroba.com*.

Se pinchará sobre el icono remarcado de la figura 13 y se pasará a modificar el *Lienzo*, figura 14.

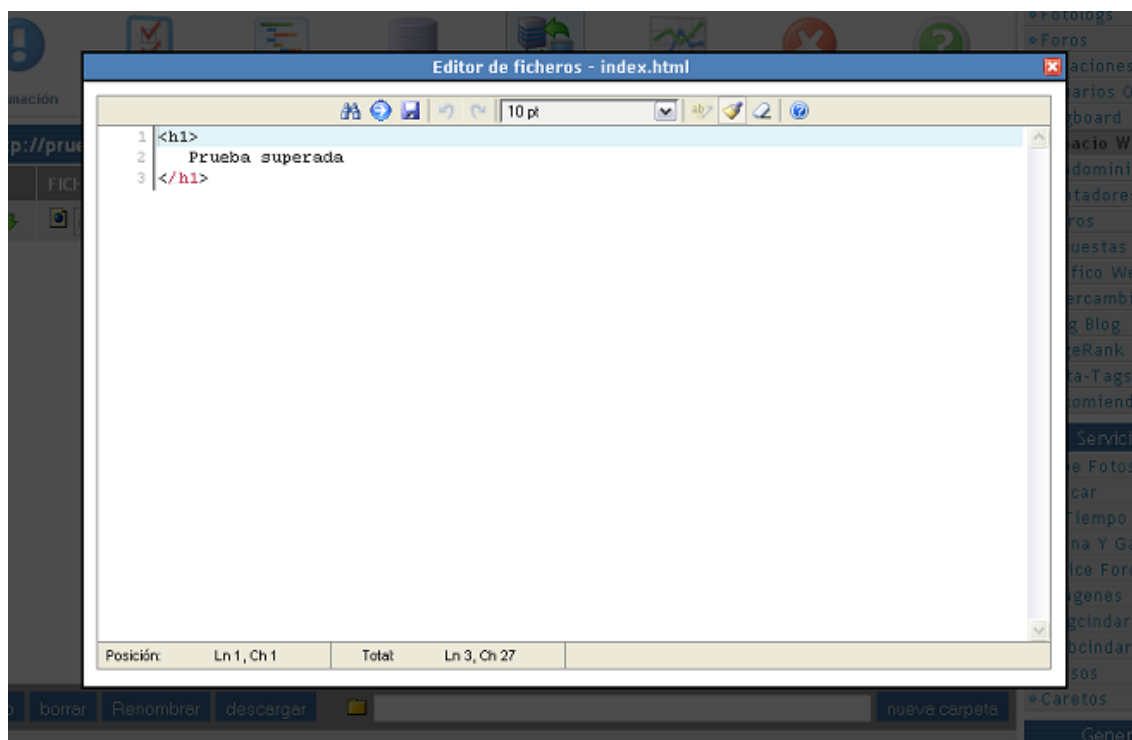


Figura 14. Editor de ficheros de *Miarroba.com*.

Una vez cambiado, se guardará y se volverá a *Facebook Developers*. Al acudir a la sección *Lienzo* se debe prestar atención al campo llamado *URL página base* que será la dirección URL mapeada de *Facebook* para la aplicación y por la que los usuarios accederán a la misma. También se debe seleccionar la *Canvas Callback URL*, que es la dirección donde *Facebook* cogerá el contenido del lienzo correspondiente a la aplicación. Un ejemplo se muestra en la figura 15.

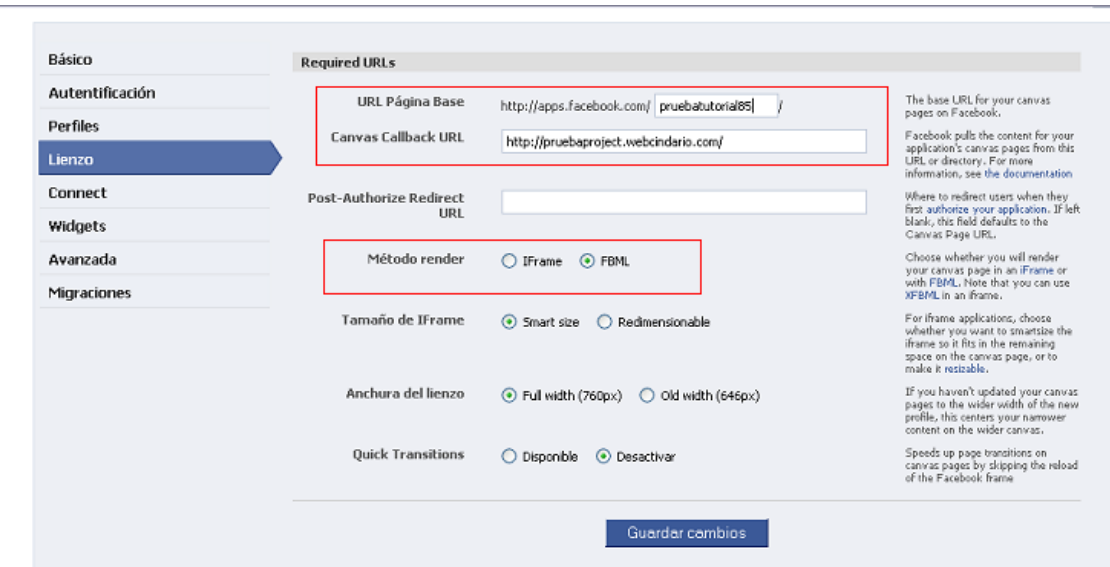


Figura 15. Apartado *Lienzo* de la configuración de la aplicación.

- URL Página Base: <http://apps.facebook.com/pruebatutorial85/>
- Canvas Callback URL: <http://pruebaproject.webcindario.com/>

Además se seleccionará la opción *FMBL*. En esta parte del tutorial no es imprescindible hacer esto, no obstante, para la realización de futuros proyectos será necesario para poder hacer uso del lenguaje *FMBL*. A continuación se guardarán los cambios.

Como resultado saldrá a la página de información de la aplicación, figura 16, donde se podrán observar todas las características de ésta así como sus estadísticas de usuarios activos mensuales, fans y el total de usuarios.

The screenshot shows the Facebook application information page for an app named 'Prueba'. The page is divided into several sections:

- Header:** 'Prueba' with a profile picture and the status 'Situación del directorio: Not Submitted'. A note says: 'Once you have completed your application, you may [enviar](#) to the Application Directory.'
- Statistics:** Three large numbers: 'Usuarios activos mensuales: 0', 'Application Fans: 0', and 'Total de usuarios: 0'.
- API Information:**
 - Clave API: [Redacted]
 - Application Secret: [Redacted]
 - Id. de aplicación: 109627382408068
 - Correo electrónico de contacto: pacosodp@hotmail.com
 - Support Email: pacosodp@hotmail.com
 - Canvas Callback URL: <http://pruebaproject.webcindario.com/>
 - Canvas URL: <http://apps.facebook.com/prueba-tutorial/>
 - FBML/iFrame: FBML
 - Modo desarrollo: Desactivar
 - Application Type: Página web
 - Private Install: No
 - Ejemplo de código: Get started quickly with some [Código de ejemplo!](#)
- Actions:** A list of links on the right side: 'Crea tu anuncio', 'DataStoreAdmin', 'Edit Application Profile', 'Editar configuración', 'Estadísticas', 'Reset Secret Key', 'Traducciones', and 'Ver el perfil de la aplicación'.

Figura 16. Página de la aplicación, con información y estadísticas de esta.

A continuación se ha de visitar la pestaña *Connect*, figura 17. En ésta se escribirá la URL principal a la que conectará la aplicación.

The screenshot shows the 'Configuración de Facebook Connect' page. The 'Connect' tab is selected in the left sidebar. The main content area includes:

- Connect URL:** A text input field containing <http://pruebaproject.webcindario.com/>, which is highlighted with a red box.
- Account Preview URL:** An empty text input field.
- Base Domain:** An empty text input field.
- Account Reclamation URL:** An empty text input field.
- Buttons:** 'Guardar cambios' (Save changes) at the bottom right.
- Help Text:** On the right, there is explanatory text about the Connect URL, including: 'Your Connect site's main URL', 'La imagen que aparece en el diálogo de Conectar a Facebook...', and 'Facebook publishes the content from here and displays it to the user in a Facebook Connect request...'.

Figura 17. Detalle del apartado *Connect*.



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

Y por último podemos comprobar como ha quedado la aplicación, figura 18, se puede leer la frase *Prueba superada*.



Figura 18. Resultado final.



2.8 Búsqueda de tecnología

Para la realización del software que funcionará en *Facebook* se buscó alguna herramienta o tecnología potente que facilitase dicha tarea. Para ello se indagó entre las distintas tecnologías que permitían la creación de aplicaciones de internet ricas, o también llamadas RIAs. De entre todas se destacó la tecnología *Flex de Adobe* y el entorno de desarrollo integrado que soporta dicha tecnología y será usado para llevar a cabo la tarea, será el *Adobe Flash Builder 4.0*.

La elección de la tecnología *Flex* viene determinada por el potencial que esta ofrece, ya que combina *MXML* (*Multimedia eXtensible Markup Language*), el cual se utiliza para describir el aspecto y comportamiento de la interfaz de usuario, y el lenguaje de programación orientado a objetos de *ActionScript 3.0*, el cual se utiliza para crear la lógica de clientes y de los componentes *MXML*. Asimismo, *Flex* incorpora una biblioteca de componentes muy completa con más de 100 componentes de interfaz de usuario extensibles (botones, cuadros de texto, formularios...) y de eficacia demostrada para crear aplicaciones de Internet sofisticadas (RIAs). (Adobe Developer, 2009).

Las aplicaciones de Internet sofisticadas creadas con *Flex* pueden ejecutarse en el explorador utilizando *Adobe Flash Player*, o fuera de él, en *Adobe AIR*, permitiéndoles que se ejecuten de un modo coherente en todos los exploradores importantes o en el escritorio.

Adobe Flash Player, instalado en más del 98% de los equipos conectados a Internet, originalmente fue creado para mostrar gráficos vectoriales avanzados, finalmente es utilizado para gestionar las aplicaciones más exigentes y con gran cantidad de datos mientras las lleva a cabo a velocidades de aplicaciones de escritorio.

Adobe AIR es una plataforma que ofrece la posibilidad a los desarrolladores web de usar las mismas tecnologías que usan para desarrollar sus aplicaciones web, pero para portarlas a un entorno de Escritorio. Es decir, que se podrá convertir las aplicaciones en *HTML*, *JavaScript*, *Flash*, *Flex* o *Ajax* en aplicaciones de escritorio. Para el usuario, esto significa que podrá disfrutar de aplicaciones más interactivas gracias a *Adobe AIR*.

Una vez seleccionada *Flex* como la tecnología a utilizar para implementar la aplicación, se debe seleccionar el entorno sobre el que se desarrollará esta. Finalmente se ha seleccionado *Adobe Flash Builder 4*, figura 19, este entorno de desarrollo integrado incluye los compiladores, la biblioteca de componentes y los depuradores necesarios para poder desarrollar aplicaciones *Flex* con todas las garantías. Esta potente herramienta se caracteriza por:

- Permite el desarrollo de contenido con un potente entorno de desarrollo integrado (IDE, por sus siglas en inglés) basado en *Eclipse™* que incluye editores de *MXML*, el lenguaje *ActionScript®* y *CSS*; además de colores de sintaxis, finalización de expresiones de programación, contracción del código, depuración interactiva estratificada y generación automática de códigos comunes.

- Diseño visual sofisticado mejorado que efectúe labores de diseño visual y previsualice la distribución de los distintos elementos de la interfaz de usuario.
- Visualización interactiva de datos mejorada, permitiendo crear tableros de datos y análisis interactivos de datos con sólo arrastrar un tipo de gráfico, soltarlo y vincularlo al origen de datos utilizando la biblioteca de gráficos para *Flex*.
- Aplicación de aspectos y estilos mejorada, personalizando la apariencia de una aplicación utilizando CSS y editores de propiedades gráficas. Se puede definir con rapidez las propiedades más usadas y previsualizar los resultados en la vista Diseño.
- El depurador (debugger) interactivo, el cual permite la creación de funciones para realizar paradas en la ejecución del programa y así poder comprobar el estado de la aplicación, componentes y variables en ese preciso instante.

Todas estas características que forman la herramienta permite la creación de aplicaciones que pueden ser integradas en *Facebook*, permitiendo a los usuarios alcanzar una experiencia satisfactoria basada en:

- **Compartir:** los usuarios pueden crear RIAs para compartir con sus amigos.
- **Divertirse:** Crear juegos sociales que permite a los usuarios jugar contra sus amigos.
- **Conexión:** los usuarios conectan con tus RIAs a través de *Facebook*.
- **Solución de problemas:** construyendo RIAs que favorecen el intercambio de información entre desarrolladores haciéndoles la tarea más sencilla.
- **Alcance:** llegar a millones de usuarios de *Facebook* a través del entramado social.

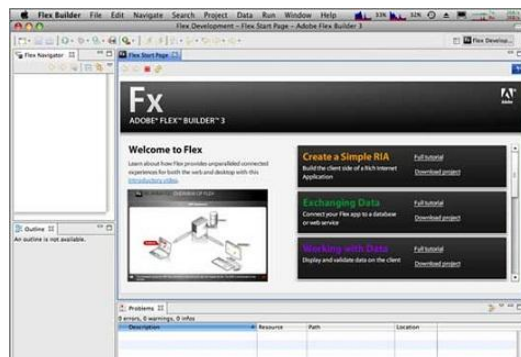


Figura 19. Detalle *Flash Builder 4*

Ahora se pasará a detallar la creación de una aplicación en Adobe Flash Builder 4.

2.8.1 Creación de aplicaciones en Adobe Flash Builder 4 para Facebook

Con el fin de explicar cómo se crean aplicaciones empleando *Adobe Flash Builder 4*, en este apartado se mostrará un ejemplo mediante los pasos que se siguen para crear una aplicación que consistirá en un navegador de fotos a través del que se podrán visionar las imágenes subidas por un usuario a *Facebook* provocando una interacción entre nuestra aplicación y dicha red social. Para que esta interacción sea llevada a cabo hay que descargarse la librería correspondiente. En este caso se tratará de la librería cliente de *ActionScript 3* para la plataforma *Facebook*, que se podrá encontrar en el enlace <http://code.google.com/p/facebook-actionscript-api/downloads/list>. (Google Code, 2011).

Como se puede observar, hay varias librerías, aunque en este caso, al tratarse de una aplicación web, ha de descargarse la versión *flex 3.4*. Si se quisiera que el software funcionase como una aplicación de escritorio debería de descargarse la versión *flex 3.4 AIRConnect*.

Ahora se pasará a configurar una nueva aplicación en *Facebook* al igual que se hizo en el ejemplo realizado en el apartado 2.7 *Introducción a Developers. Prueba*. Se le dará nombre a la aplicación y se anotará el número de la API y su clave secreta. A continuación se irá a la pestaña *Avanzada* y en la opción *Application Type* se seleccionará *Desktop*, figura 20, ya que para la realización de pruebas, así como para usar el modo *debug* de *AFB4* (*Adobe Flash Builder 4*), es necesario hacer esto antes de subir la aplicación a un servidor.

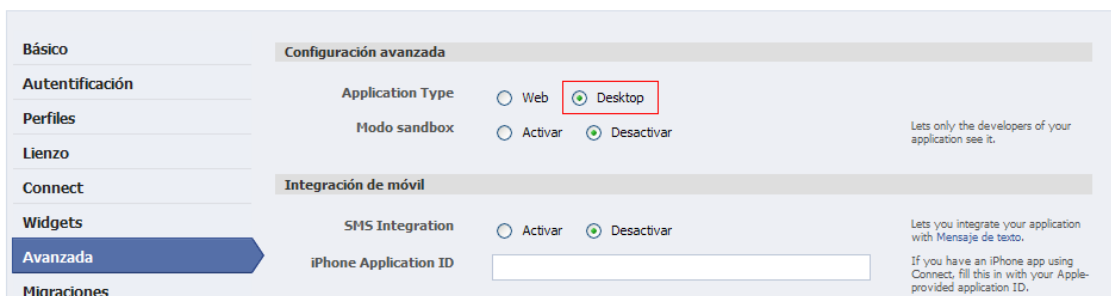


Figura 20. Seleccionar *Desktop*. Pestaña avanzada.

Posteriormente se hará clic en *Guardar cambios* para almacenar la nueva configuración y se ejecutará *AFB4*.

Lo primero será crear un nuevo proyecto, figura 21, por lo que se debe de seleccionar *File > New > Flex Project*).

Se abrirá el cuadro para configurar la aplicación, donde se le dará nombre y donde se ha de seleccionar que la aplicación será Web.

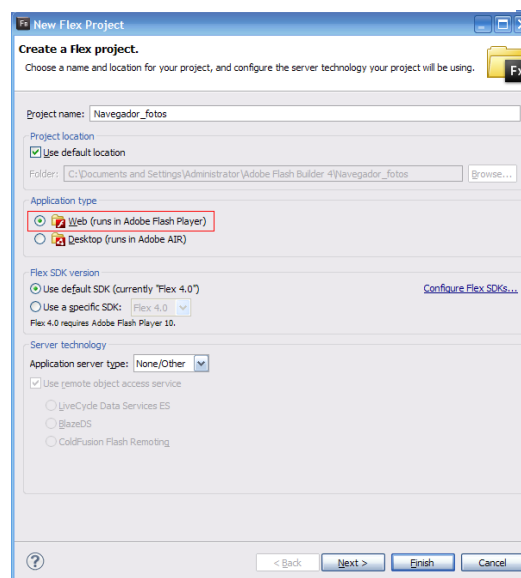


Figura 21. Proyecto Web nuevo.

Lo siguiente será hacer Clic en *Finish*.

Después, aparecerá la vista general del programa. En la esquina superior izquierda de la pantalla se observará una ventana llamada *Package Explorer*, figura 22, vista en la que aparecerán todos los proyectos con sus componentes. Es aquí donde se añadirá la librería anteriormente descargada. Se arrastrará el archivo *.swc* a la carpeta *libs* del proyecto en curso.

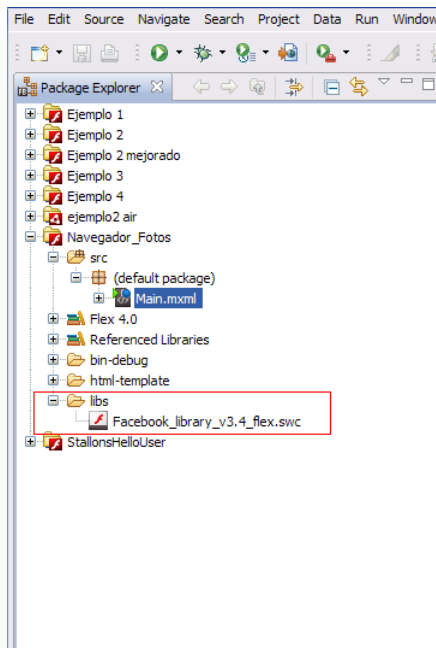


Figura 22. Package Explorer de AFB4.

➤ Creación de la interfaz del programa.

Pinchar sobre el botón de diseño situado en la parte superior de la ventana de código. Ahora se puede seleccionar de la ventana *components* las etiquetas, botones, ventanas con listas, etc.

Como se observa en la figura 23, se ha arrastrado a la hoja de diseño un botón y se le ha etiquetado bajo el nombre de *1. Login to Facebook*. En el campo *On Click* se seleccionará la acción a realizar cuando se pulse dicho botón. En este caso se optará por la acción *doLogin()* que más tarde se detallará. También se pueden modificar otros campos de estilo y de apariencia.

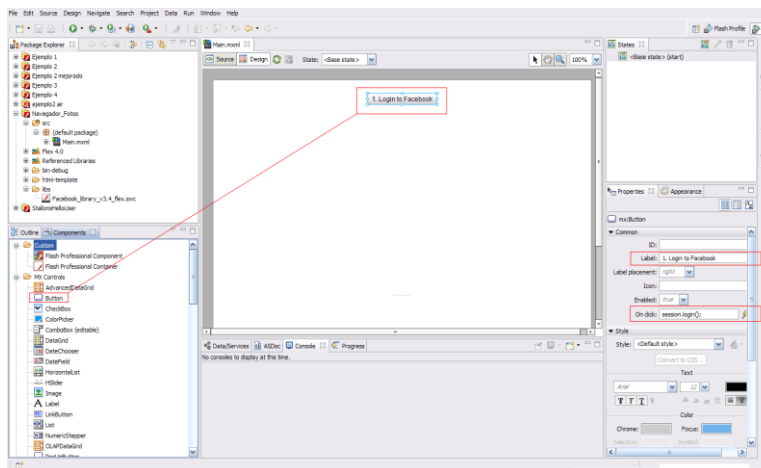


Figura 23. Vista general AFB4.

Se realizará lo mismo para otro botón más, el cual obtendrá la lista con los álbumes del usuario:

- Label = "2. Browse Photos"; onClick = getPhotoAlbums().

Una vez realizado esto se introducirán dos ventanas. Una contendrá la lista con los álbumes del usuario y en la otra se podrán visionar las fotos en miniatura correspondientes a un determinado álbum a modo de lista. También se incluirá un visor de imagen. Figura 24.

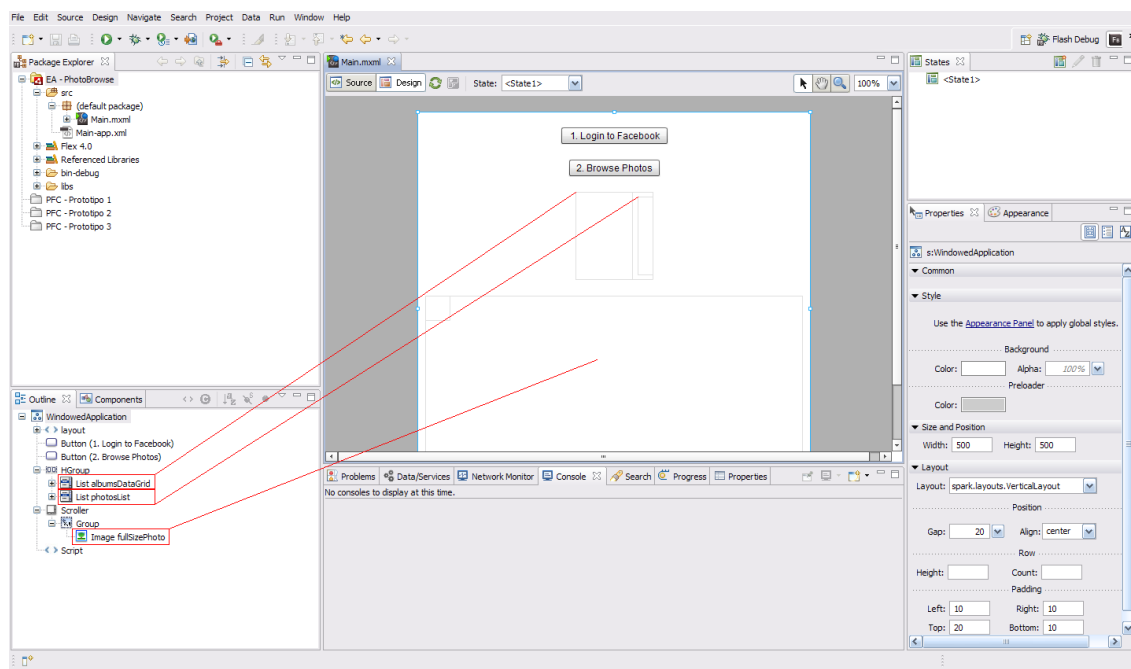


Figura 24. Vista diseño AFB4.

De modo que un usuario podrá elegir un álbum de la lista, automáticamente se mostrarán todas las fotos en miniatura del álbum seleccionado y, finalmente podrá seleccionar una imagen y verla en su tamaño original en el visor de imagen.

A continuación se pasará a detallar las propiedades necesarias para cada una de las dos listas utilizando las ventanas que se encuentran a la derecha de la vista de diseño como se observa en la figura 25.

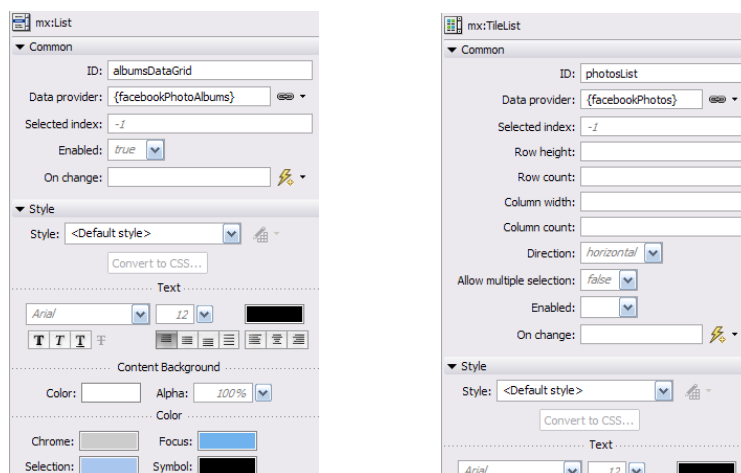


Figura 25. Ventana propiedades de cada lista insertada.

A cada lista se le proporcionará un identificador (*albumsDataGrid* y *photosList*, respectivamente) y un proveedor de datos, el cual será el suministrador de la información que solicite la lista. La primera lista corresponderá con los álbumes del usuario, por lo que a su proveedor de datos se le llamará *facebookPhotoAlbums*. La segunda lista corresponderá con las fotos de un álbum determinado y seleccionado de la lista anteriormente descrita. Su proveedor de datos será nombrado como *facebookPhotos*.

A continuación, se hará clic sobre el botón *category view*, el cual viene remarcado en la figura 26 y 27, este botón cambia la visualización de las propiedades de los componentes. Permitirá observar todas las propiedades de un componente ordenadas por categorías. Es igual a la anterior vista de propiedades de un componente, figura 25, pero esta es mucho más específica.

Para la lista con los álbumes se seleccionará la propiedad *Events > Click* y la completaremos con el método *showSelectedAlbum()*, el cual indica que cuando se seleccione un álbum mediante un clic de la lista, se muestre en la otra ventana todas las imágenes en miniatura que contiene dicho álbum, figura 26.

Para la segunda ventana se selecciona de nuevo *Events > Click*, se completará el campo igualando el visor de imagen a la foto en miniatura seleccionada. Esto quiere decir, que cada vez que se seleccione una foto en miniatura de la lista, se mostrará la imagen en su tamaño original, figura 27.

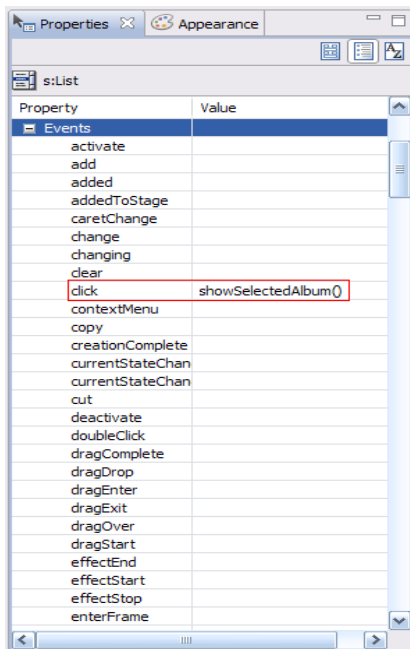


Figura 26. Lista álbumes.

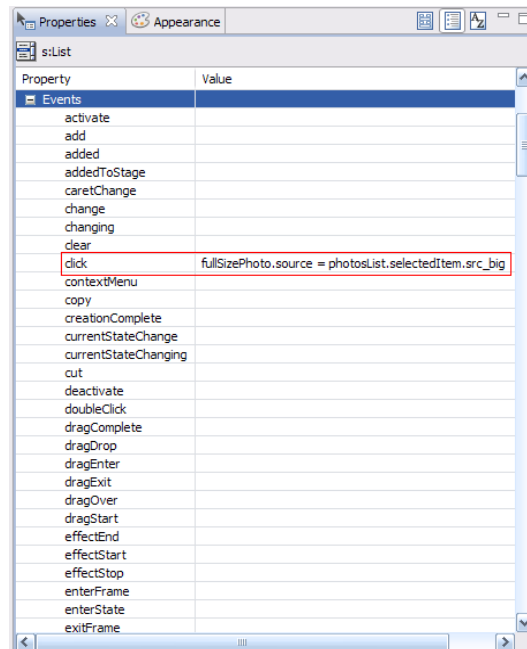


Figura 27. Lista fotos.

La imagen a tamaño original será mostrada en el visor de imagen, este será identificado por el nombre de *fullSizePhoto*.



Ahora dar clic sobre el botón de *source* para observar el código creado, figura 28. Ahora el *main.mxml* recoge todo lo realizado anteriormente en líneas de código:

```
<s:Button label="1. Login to Facebook" click="doLogin()" />
<mx:Button label="2. Browse Photos" click="getPhotoAlbums()" />

<s:HGroup verticalAlign="middle" >
    <s>List id="albumsDataGrid" dataProvider="{facebookPhotoAlbums}"
        click="showSelectedAlbum()" minHeight="100">
        <s:itemRenderer>
            <fx:Component>
                <s:ItemRenderer>
                    <s:Label text="{data.name}" padding="5"
                        paddingLeft="5" paddingBottom="5" paddingRight="5"
                        buttonMode="true" />
                </s:ItemRenderer>
            </fx:Component>
        </s:itemRenderer>
    </s>List>

    <s>List id="photosList" dataProvider="{facebookPhotos}"
        click="fullSizePhoto.source = photosList.selectedItem.src_big"
        minHeight="100">
        <s:layout>
            <s:TileLayout verticalAlign="top"/>
        </s:layout>
        <s:itemRenderer>
            <fx:Component>
                <s:ItemRenderer>
                    <mx:Image source="{data.src_small}" buttonMode="true"/>
                </s:ItemRenderer>
            </fx:Component>
        </s:itemRenderer>
    </s>List>

</s:HGroup>

<s:Scroller width="100%" height="100%">
    <s:Group>
        <mx:Image id="fullSizePhoto"/>
    </s:Group>
</s:Scroller>

</s:WindowedApplication>
```

Figura 28. Código generado automáticamente.

Ya está creada la interfaz del programa ahora se describirán las clases y variables del programa.

➤ Definición de clases y variables.

Primero, se creará un bloque de script donde se definirán las variables y las funciones de la aplicación, figura 29, este bloque está situado entre el inicio de la aplicación y la definición de la interfaz.

```
<mx:Script>
    <![CDATA[
    ]]>
</mx:Script>
```

Figura 29. Bloque de Script.



Dentro de este bloque de script, se importará la clase *FacebookDesktop*, figura 30, la cual permitirá llamar a los métodos que permiten tanto la conexión como las llamadas *FQL*. Esta clase viene recogida en la librería anteriormente agregada a la carpeta *libs* de nuestro proyecto. (Google Code, 2010).

También se añadirá la clase *ArrayCollection*, figura 30, perteneciente a *MXML*, la cual permite el manejo de Arrays.

Las variables a crear son:

- *facebookPhotoAlbums*: Array donde se almacenará toda la información perteneciente a un álbum.
- *facebookPhotos*: otro array donde se almacenarán las fotos de un álbum en concreto.
- *appID*: string donde se almacena la ID de la aplicación suministrada por *Facebook*.

```
import com.facebook.graph.FacebookDesktop;
import mx.collections.ArrayCollection;

private var appID:String = "*****";

private var facebookPhotoAlbums:ArrayCollection;
private var facebookPhotos:ArrayCollection;
```

Figura 30. Clases y variables de la aplicación.

Ahora se definirán métodos y funciones del software.

➤ Creación de la sesión de conexión con *Facebook*.

Al iniciar la aplicación, está comenzará cargándose, para ello lanzará la función *init()*. En esta función se realiza la conexión entre el servidor de *Facebook* y la aplicación, verificando que esta está dada de alta en *FacebookDevelopers* mediante la ID de la aplicación, figura 31. Para realizar esta verificación utilizará el método, también llamado *init*, perteneciente a la clase anteriormente descrita *FacebookDesktop*, se le pasarán dos parámetros, la ID de la aplicación y una función a la que se la llamará una vez completada la conexión, *handleLogin()*.

```
private function init():void {
    FacebookDesktop.init(appID, handleLogin);
}
```

Figura 31. Función *init()*.

Ahora el usuario de la aplicación podrá conectarse a la aplicación, para ello debe conectarse a *Facebook*. Pulsando el botón *1. Login to Facebook*, se ejecutará la función asociada al botón anteriormente especificada, *doLogin()*, figura 32. Se ejecutará el método *login*, perteneciente a la clase *FacebookDesktop*, que se le pasarán dos parámetros:



- La función a la que se devolverá la llamada una vez completada la conexión, *handleLogin()*.
- Los permisos necesarios para el correcto funcionamiento de la aplicación, en este caso serán los permisos relacionados con las fotos.

```
private function doLogin():void {  
    FacebookDesktop.login(handleLogin, ['user_photos']);  
}
```

Figura 32. Función *doLogin()*.

Esta función abrirá una ventana pop-up permitiendo al usuario conectarse a *Facebook*.

Una vez que se haya completado la conexión, tanto del usuario como de la aplicación, se llamará a la función *handleLogin()*, figura 33. Comprobará que se ha devuelto el objeto *session*, el cual contiene información respecto a la conexión y al usuario de la aplicación. En la barra de estado de la aplicación aparecerá un mensaje indicando que la conexión ha sido satisfactoria.

```
private function handleLogin(session:Object, fail:Object):void {  
    if (session != null) {  
        status = "Successfully logged in";  
    }  
}
```

Figura 33. Función *handleLogin()*.

➤ Obtención de álbumes y mostrar fotos.

La aplicación está ahora preparada para poder ser usada. Se podrá pulsar el botón 2. *Browse Photos*, ejecutándose la función asociada a dicho botón, *getPhotoAlbums()*, figura 34. En esta función se realizará la primera consulta, se almacenará en un string llamado *fql* la consulta, seleccionando el nombre y el número de identificación del álbum del usuario conectado a la aplicación, la consulta se realiza sobre la tabla *álbum*, perteneciente a la base de datos de *Facebook*. Se utilizará el método *fqlQuery* perteneciente a la clase *FacebookDesktop*, pasándole como primer parámetro el string *fql* con la consulta, y el segundo será una función a la que llame una vez se haya completado de recibir la información correspondiente a la consulta.

```
private function getPhotoAlbums():void {  
    var fql:String = "select name, aid from album where owner =me()";  
    FacebookDesktop.fqlQuery(fql, handleGetAlbumsResponse);  
}
```

Figura 34. Función *getPhotoAlbums()*.

Al ejecutarse la función *handleGetAlbumsResponse()*, figura 35, se comprobará que se ha recibido información y se almacenará en la variable global *facebookPhotoAlbums* como un *ArrayCollection*. Esto provocará que la lista de álbumes, *AlbumsDataGrid*, muestre el nombre de los álbumes del usuario, debido a que su proveedor de datos, *facebookPhotoAlbums*, le está suministrando dicha información.



```
private function handleGetAlbumsResponse(albums:Object, fail:Object):void {  
    if (albums != null)  
        facebookPhotoAlbums = new ArrayCollection(albums as Array);  
    else  
        status = "Error";  
}
```

Figura 35. Función *handleGetAlbumsResponse()*.

Ahora se podrá seleccionar un álbum de la lista, al hacer esto se mostrará en la lista de imágenes las fotos en miniatura del álbum seleccionado. Esto se produce debido que al seleccionar un álbum, se dispara la función *showSelectedAlbum()*, figura 36. En esta función se realiza otra consulta FQL, se pedirán las fotos en tamaño grande y pequeño de álbum seleccionado. La consulta se realizará sobre la tabla *photo*, perteneciente a la base de datos alojada en *Facebook*. El procedimiento para realizar la consulta es igual siempre, en este caso la función a la que se llamará una vez acabada la consulta es *handleGetPhotosResponse()*.

```
private function showSelectedAlbum():void {  
    var fql:String = "select src_big, src_small from photo  
    where aid = '" + albumsDataGrid.selectedItem.aid + "'";  
    FacebookDesktop.fqlQuery(fql, handleGetPhotosResponse);  
}
```

Figura 36. Función *getPhotoAlbums()*.

La función *handleGetPhotosResponse()*, figura 37, será la encargada de comprobar que la consulta se ha llevado a cabo de manera satisfactoria y almacenará la información recibida a modo de Array en la variable global *facebookPhotos*, esta variable es el suministrador de datos de la lista de imágenes en miniatura, por lo que finalmente se podrá observar todas las imágenes pertenecientes a un álbum en la lista *photoList*.

```
private function handleGetPhotosResponse(photos:Object, fail:Object):void {  
    if (photos != null)  
        facebookPhotos = new ArrayCollection(photos as Array);  
    else  
        status = "Error";  
}
```

Figura 37. Función *handleGetPhotosResponse()*.

Ahora el usuario puede seleccionar una imagen de entre todas las que hay, al realizar esto, el visor de imagen mostrará la imagen seleccionada en tamaño original. Esto es debido a que en la definición de la lista *photoList*, se ha declarado que al realizar clic sobre una imagen, el visor *fullSizePhoto* tome el valor de la imagen seleccionada, pero en su tamaño original, figura 38.

```
click="fullSizePhoto.source = photosList.selectedItem.src_big"
```

Figura 38. Sentencia perteneciente a la lista *photoList*.

A continuación se realizará una ejecución de prueba de la aplicación.

2.8.2 Funcionamiento del Programa

Ahora se pasará a detallar el uso del programa.

1. Identificarse en Facebook

Se pulsará el botón *1. Login to Facebook* para que se abra automáticamente una ventana que permitirá identificar al usuario en *Facebook*, figura 39, mediante su nombre de usuario y contraseña. Si es la primera vez que el usuario ejecuta esta aplicación, *Facebook* requerirá que acepte los permisos necesarios para que la aplicación pueda acceder a la información sobre las fotos.

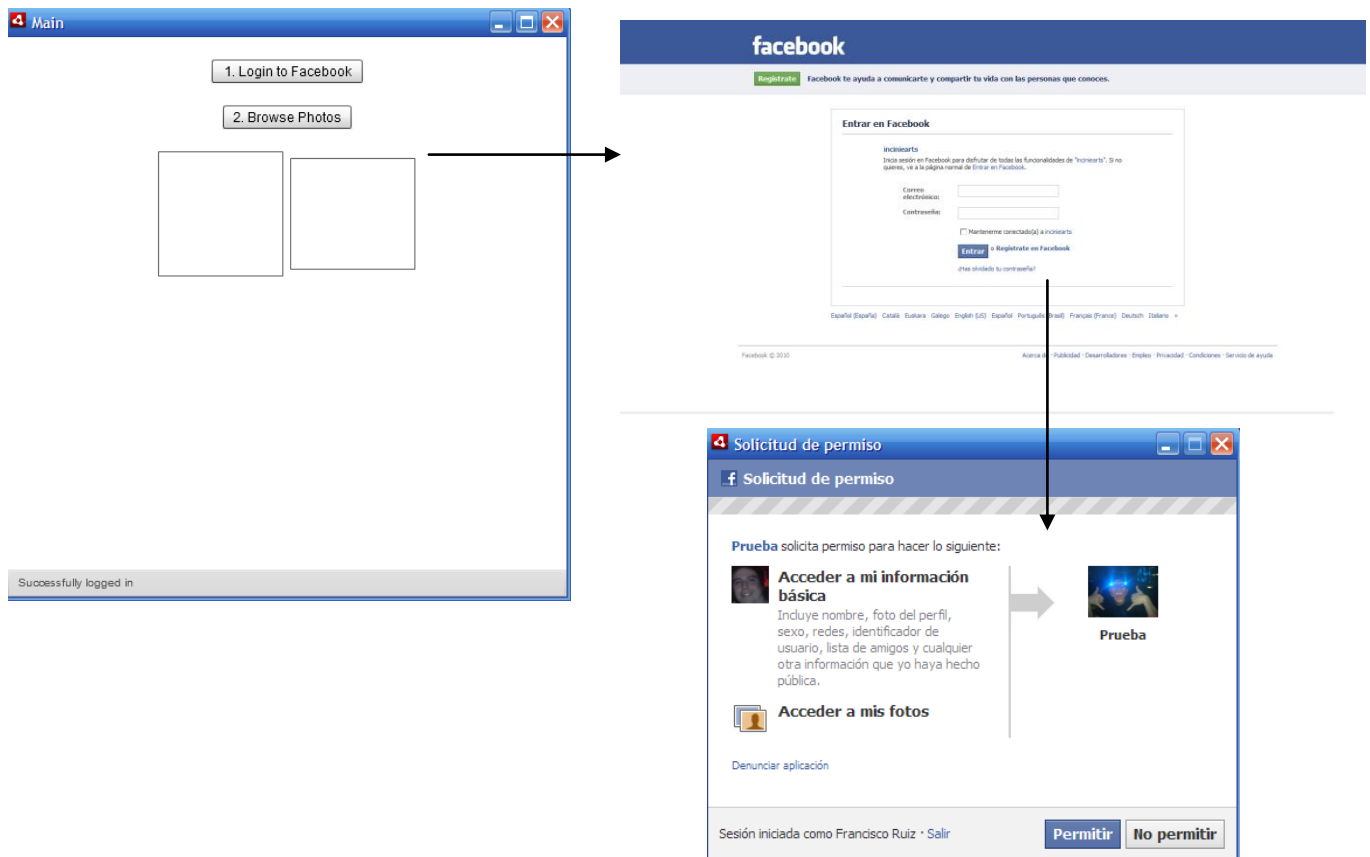


Figura 39. Login en *Facebook*.

2. Navegador de Fotos

A continuación se pulsará el botón 2. *Browse Photos*. Al hacer esto el programa cargará los álbumes mostrando su nombre en el cuadro de la izquierda. En dicho cuadro será donde se podrá seleccionar el álbum que se quiera, llegando a visionar sus imágenes en miniatura en el cuadro de la derecha, figura 40. En este mismo cuadro, si se selecciona una imagen, ésta será mostrada a tamaño completo en la parte inferior.

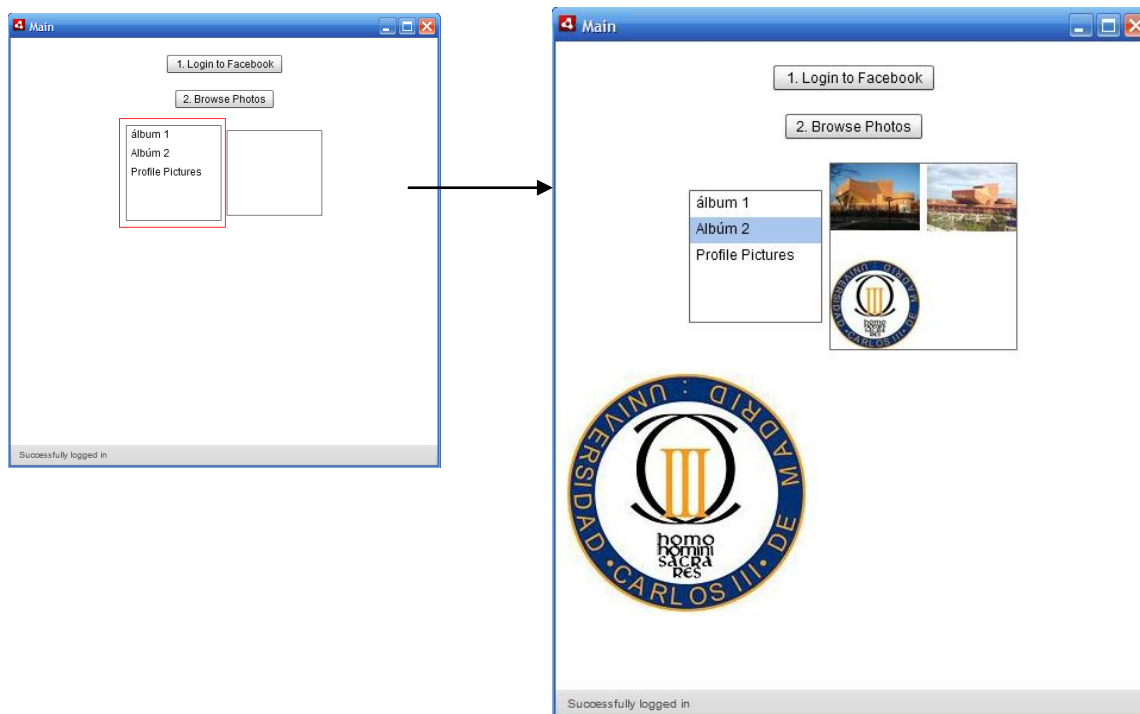


Figura 40. Visualización de los álbumes y sus miniaturas.



2.11 Conclusiones

Como se ha visto, las redes sociales son ahora un mundo a explotar con un mercado potencial de usuarios enorme que crece a cada día que pasa, y que permite que cientos de empresas y desarrolladores creen utilidades que puedan ser usados por estos a través de la red social, haciendo que la experiencia del usuario con su red sea elevadísima.

Una vez concluida la lectura de este capítulo, uno ya está preparado para realizar aplicaciones ricas, RIAs, que puedan ser utilizadas dentro de *Facebook* utilizando la tecnología que *Adobe* pone a disposición, *Adobe Flash Builder 4*.

Siempre que se entra en contacto con una nueva tecnología cuesta un poco hacerse con ella, pero una vez que se dominan todos los recursos de ésta uno puede llegar a hacer grandes trabajos con estos kits de desarrollo y con un aspecto visual inmejorable.

La máxima complejidad que se puede presentar a la hora de aprender *Flex*, que como se sabe es el lenguaje informático utilizado para realizar los programas en *Adobe Flash Builder*, es que este lenguaje es la unión de otros semejantes para que se sumen todas las cualidades de cada uno de éstos, haciendo que el resultado sea un lenguaje muy potente que ofrece al desarrollador la capacidad de crear buenas RIAs.

Es tremendamente interesante el potencial y capacidad de esta tecnología aprendida, y no sólo para la creación de aplicaciones para *Facebook*, ya que, finalmente, parece que la única limitación que puede presentar dicha tecnología es la que puede llegar a ofrecer el desarrollador.



3. ESTUDIO DE VIABILIDAD DEL SISTEMA

En este apartado se llevará a cabo el estudio de viabilidad del sistema para estudiar la posibilidad de realizar una aplicación que tenga conectividad con las BB.DD de la red social *Facebook*.

3.1 Introducción

Se trata del análisis de un conjunto concreto de necesidades para proponer una solución a corto plazo que tenga en cuenta las restricciones económicas, técnicas, legales y operativas.

3.1.2 Propósito del plan.

El objetivo de un *Estudio de Viabilidad del Sistema* es realizar un estudio de las necesidades planteadas por el cliente o usuario realizando una descripción general de la misma. (Portal de Administración Electrónica, 2011). Para ello se determinan los objetivos, se inicia el estudio de los requisitos y se identifican las unidades organizativas afectadas estableciendo su estructura. Además se analizan las posibles restricciones, tanto generales como específicas, que puedan condicionar el estudio y la planificación de las alternativas de solución que se propongan.

En este caso, basándose en las necesidades del cliente, se estudiará primero el alcance del sistema realizando una identificación de los primeros requisitos, una descripción del nuevo sistema y una planificación sin detenerse demasiado en el estudio de viabilidad del sistema debido a que el riesgo técnico es bajo, ya que no se esperan problemas legales y no existen muchas alternativas posibles.

Con el fin de explicar lo anteriormente indicado en la primera parte del capítulo se detallará la composición del equipo de trabajo necesario para el proceso y la planificación. Finalmente, con el fin de facilitar la implicación activa de los usuarios en la definición del sistema, se identifican sus perfiles, dejando claras sus tareas y responsabilidades.

En un segundo bloque, se realizará un estudio de la situación actual en la que se encuentran los sistemas de información existentes en el momento que se inicia su estudio. Como resultado de esta actividad se genera un diagnóstico estimando la eficiencia de los sistemas de información existentes e identificando los posibles problemas y mejoras.

Posteriormente, se realizará la definición de los requisitos que el sistema debe cumplir mediante una serie de sesiones de trabajo con los usuarios participantes. Una vez finalizadas la información extraída es analizada y añadida al catálogo de requisitos. Estos últimos serán detallados, completos y no ambiguos con el fin de crear una buena base para futuras etapas del ciclo de vida del proyecto. El lenguaje en el que serán descritos será sencillo con el fin de que el cliente pueda comprenderlos y validarlos sin problema.



Por último lugar, la solución final elegida será expuesta para ser desarrollada por el equipo de trabajo.

3.2 Establecimiento del alcance del sistema

En esta actividad se estudia el alcance de la necesidad planteada por el cliente realizando una descripción general de la misma. Se determinan los objetivos, se inicia el estudio de requisitos y se identifican las unidades organizativas afectadas estableciendo su estructura. También, se analizan las posibles restricciones que puedan condicionar el estudio y la planificación de las alternativas de solución que se proponen. Finalmente, se detalla la composición del equipo de trabajo y se identifican los perfiles de los usuarios en la definición del sistema.

3.2.1 Estudio de la solicitud

En la solicitud del cliente se especifica la necesidad de desarrollar una aplicación para la red social *Facebook*. Dicha aplicación será un juego que deberá tener conectividad con las bases de datos de dicha red social, de manera que los datos obtenidos puedan ser empleados durante el desarrollo de la aplicación.

La aplicación a desarrollar ha de cumplir una serie de características propuestas por el cliente:

- Permitir que un usuario de la red social pueda conectarse a la aplicación.
- Interaccionar con las bases de datos de *Facebook* de manera que dichos datos se utilicen durante el desarrollo del juego.
- La aplicación debe permitir desarrollar una partida similar a las ofrecidas por el juego de mesa “¿Quién es quién?”.
- Las tarjetas del juego, compuestas por personas con distintas características físicas en el juego original, deberán ser sustituidas en la aplicación por amigos/as del usuario conectado al juego. Además se deberá poder visualizar los datos personales de éstos.
- La interfaz gráfica deberá ser intuitiva con el fin de que el juego pueda desarrollarse de manera fluida pudiéndose realizar el intercambio de preguntas y respuestas con la máquina así como la visualización de los datos personales de amigos/as que participan indirectamente en el juego (a modo de tarjetas como se ha podido observar en el punto anterior).
- Emplear una inteligencia artificial (IA) definida con la que el usuario pueda enfrentarse en partidas totalmente disputadas. Esta estará alojada dentro de la propia aplicación.

En lo referente a costes de almacenamiento, el cliente prefiere la utilización de servidores gratuitos para alojar la aplicación de modo que se pueda ahorrar en gastos y



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

en mantenimiento. Por este motivo se ha de obtener un alojamiento en internet que ofrezca las suficientes garantías para ello.

Por último, el cliente, además de poder conectarse a la aplicación a través de la dirección web ofrecida por el servidor gratuito, ha mostrado su deseo de que la aplicación esté totalmente integrada en *Facebook*, de manera que uno pueda acceder al juego directamente desde la red social y jugar dentro de ésta.

En cuanto a las personas encargadas de llevar a cabo este proyecto, pueden distinguirse dos: Jessica Rivero Espinosa, como directora del mismo, y Francisco Ruiz Gualda, como desarrollador.



3.2.2 Identificación del alcance del sistema

El presente proyecto consta de un único módulo a desarrollar cuyos requisitos serán identificados en este apartado, identificando las principales funcionalidades que se deberán desarrollar.

Principalmente, la aplicación debe cumplir una serie de características de forma que ésta sea similar al juego de mesa “¿Quién es quién?”, pudiendo alcanzar una experiencia de juego parecida. Las principales características a cumplir son:

- El juego se compone de dos tableros formados por un número limitado de amigos/as del usuario conectado a la aplicación elegidos aleatoriamente. Cada uno de estos tableros pertenece a uno de los participantes del juego: usuario y máquina.
- Cada uno de los participantes tendrá una tarjeta con un amigo/a seleccionado al azar que será el objetivo a descubrir por su oponente.
- Intercambio de preguntas y respuestas, definidas previamente en el sistema, entre los participantes del juego.
- Los jugadores han de poder descartar a amigos/as con el fin de llegar a la solución final siempre en función de las preguntas y respuestas realizadas entre ellos.
- La aplicación permitirá la visualización de los datos personales de los amigos/as que conforman el tablero y de los que son objetivos.
- La inteligencia artificial deberá contestar con total exactitud a las preguntas formuladas por el usuario, así como, realizar el correcto descarte de amigos/as del tablero en función de la respuesta dada por el usuario a la pregunta realizada por ella misma. Ésta también será capaz de determinar cuándo ha llegado a una solución final y, por tanto, resolver la partida.

Las preguntas que se podrán intercambiar entre los participantes del juego serán definidas antes del desarrollo del software mediante una serie de reuniones mantenidas con el cliente.

Como todo juego, se creará un apartado de reglas e instrucciones de manera que el usuario que acceda a la aplicación no tenga dudas del desarrollo de la misma. Estas reglas e instrucciones vendrán recogidas en una página web alojada en el mismo servidor que la aplicación.

Independientemente del desarrollo del juego la aplicación debe ofrecer la posibilidad de desconectarse en cualquier instante.



3.2.3 Identificación de los interesados en el sistema (stakeholders)

En este punto se lleva a cabo una identificación de todas aquellas personas o entidades interesadas de alguna forma en el proyecto. En concreto, en esta aplicación se encunetran las siguientes entidades:

- Cliente: Es la persona que realiza la solicitud de desarrollo del sistema. En este caso la cliente es Jessica Rivero Espinosa, profesora perteneciente al departamento LaBDA (Laboratorio de Bases de Datos Avanzadas) de la universidad Carlos III de Madrid.
- Usuarios finales: Son aquellas personas que podrán disfrutar de la aplicación. Es decir todas aquellas personas registradas en *Facebook*.
- Tutora y coordinadora del proyecto: Jessica Rivero Espinosa, dirigirá, organizará y orientará al autor del proyecto, además realizará tareas de supervisión a lo largo de toda la realización del proyecto.
- Autor del proyecto: Francisco Ruiz Gualda. Es la persona encargada de la realización del proyecto, siendo además el máximo responsable del mismo y del producto generado.



3.3 Estudio de la situación actual

En este apartado se analiza la situación actual de las aplicaciones existentes que han sido desarrolladas para funcionar bajo la red social *Facebook*.

3.3.1 Valoración del estudio de la situación actual

En la actualidad, los usuarios de *Facebook* instalan 20 millones de aplicaciones por día. (PC World, 2009) Por lo que el mercado de aplicaciones es muy amplio y con una alta competencia saliendo a la luz miles de aplicaciones al día, además de actualizaciones y traducciones a otros idiomas de las ya existentes.

Debido al número tan elevado de aplicaciones, es muy difícil llegar a hacer una catalogación de éstas de modo que se tengan todas en cuenta. El género por excelencia, el más utilizado a la par que el más abundante, es el de los juegos sociales, donde se enmarcará la aplicación que dará lugar este proyecto. Centrándose en este género se pueden encontrar:

- **Juegos sociales online masivos:** aquellos en los que el usuario tiene que evolucionar diariamente una ciudad, persona o granja mediante la utilización de unos recursos y con la ayuda de tus amigos. *CityVille*, *Sims* y *FarmVille* están entre las 10 aplicaciones más utilizadas con cerca de 150 millones de usuarios diarios. (Hodge, 2008). Estos juegos son altamente adictivos y se aprovechan de que el usuario ha de ser muy activo con la aplicación si desea evolucionar rápidamente.
- **Juegos online:** no se requiere una participación diaria del usuario para obtener una evolución correcta del juego. Éstos suelen ser juegos más parecidos a los tradicionales (Juegos de cartas, mini juegos, *quiz*, etc.). También precisan de un servidor potente, aunque la media de usuarios es más baja que la del anterior tipo. La complejidad de desarrollo puede ser elevada, variando desde los juegos de póker hasta el típico juego pregunta/respuesta, pudiendo ser, en algunos casos, semejante a la del anterior tipo.

La diferencia entre ambos tipos es mínima aunque en la práctica hace que la experiencia del usuario sea totalmente diferente.

La complejidad en el desarrollo de estos juegos depende de la ambición de los mismos. Un juego del tipo *Farmville* o uno de póker ha de contar con un servidor potente debido al gran número de usuarios, así como de una base de datos de gran capacidad por la cantidad de información a almacenar. Su desarrollo es de una complejidad alta, por lo que es necesario un equipo amplio de desarrolladores especializados en distintas ramas: programadores, diseñadores 3D para modelos, diseñadores 2D para texturas, diseñadores de efectos sonoros, músicos, especialistas en redes, etc.

En cuanto a las tecnologías utilizadas, el estándar es c++, pues para desarrollar un proyecto de estas características son necesarios varios motores y *engines* (gráfico,



físico, de red, inteligencia artificial, etc.), y la mayoría de los que pueden ser adaptados están desarrollados en este lenguaje. Por lo que en estos proyectos suele ser recurrente el comprar *engines* ya desarrollados antes que realizarlos directamente, ahorrando tiempo, ya que si una empresa asume todas las labores de desarrollo de un proyecto de esta envergadura, el producto final podría tardar varios años en salir a la luz.

3.3.2 Realización del diagnóstico actual

Una vez realizado el estudio de las aplicaciones ya existentes se puede concluir que el proyecto que se va a llevar a cabo es un juego social de tipo online, es decir, no es necesaria una total involucración por parte del usuario para sacarle el máximo rendimiento a la aplicación.

Debido a que la disponibilidad de recursos se limita a un único programador, se ha realizado una búsqueda de una tecnología que pudiese agrupar todos los módulos implicados (IA, interfaz gráfica, red, etc.) simplificando la tarea de dicho programador.

Finalmente, la tecnología seleccionada es *Flex* de *Adobe*, pues está específicamente diseñada para crear RIAs. Esta tecnología une *ActionScript*, empleado para desarrollar la IA, métodos, funcionalidades, etc. de la aplicación, y *MXML*, que permite desarrollar toda la interfaz gráfica.

Se utilizará un servidor gratuito y fiable para alojar la aplicación, de manera que se ahorren gastos lo máximo posible. El elegido es *000WebHosting.com*, avalado por las buenas críticas de sus usuarios.

Finalmente, el punto fuerte de este proyecto puede que resida en la originalidad de la propuesta, ya que, aunque esté basado en un juego de mesa de largo recorrido, no abundan en *Facebook* juegos de estas características.

3.4 Definición de los requisitos del sistema

Una vez realizado el estudio de la solicitud del cliente se realiza la extracción de requisitos con el fin de cumplir con las funcionalidades que debe proporcionar la aplicación.

Esta extracción de requisitos se realiza también con el objetivo de orientar al cliente permitiendo al equipo de desarrollo obtener una lista de requisitos detallada, completa y sin ambigüedades. Estos requisitos proporcionarán una visión general de la aplicación, sin ahondar en aspectos técnicos, estableciendo las principales funcionalidades y restricciones, y sirviendo de base a posteriores procesos del ciclo de vida del software.

3.4.1 Especificación de los casos de uso

Antes de la redacción de los requisitos se han de identificar los casos de uso con el fin de lograr una mejor definición de estos. Para ello, en la figura 41 se muestra el diagrama de casos de uso del sistema que se va a desarrollar.

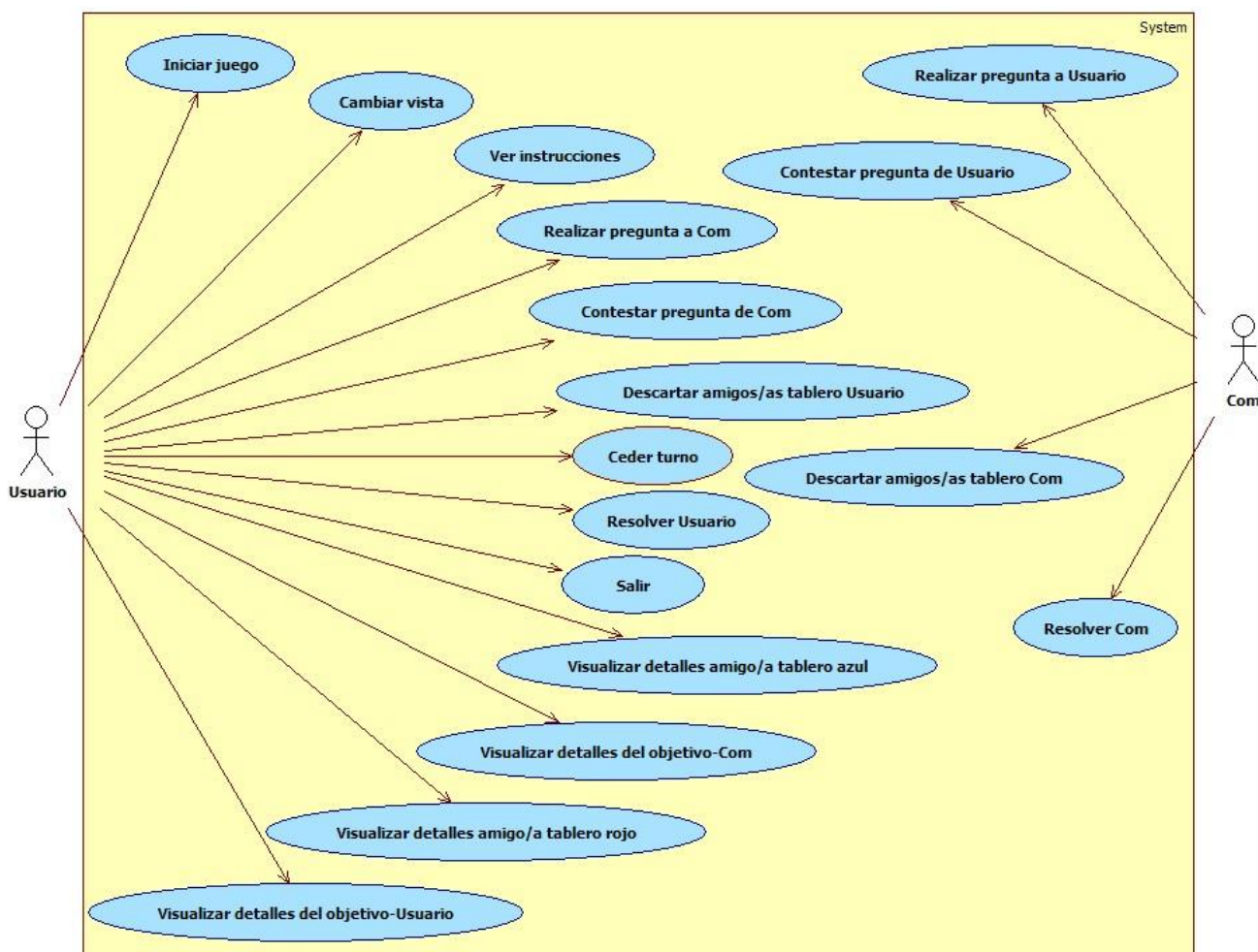


Figura 41. Diagrama casos de uso.



3.4.1.1 Especificación detallada de los actores

Se llama actor a toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos, además de entidades abstractas, como el tiempo. (Wikipedia, 2011)

Una vez aclarado que se entiende por actor, se pasan a describir los atributos de los actores que interactúan con el sistema en el diagrama de casos de uso de la figura 41 y que se definirán en la Tabla 1 y Tabla 2:

- **Actor:** Nombre del Actor.
- **Casos de uso:** Nombre de los casos de uso en los que participa.
- **Tipo:** Primario (la comunicación la inicia el actor) o secundario (la comunicación la inicia el sistema).
- **Descripción:** Breve descripción del autor.

Actor	Usuario
Casos de uso	Iniciar juego, Cambiar vista, Ver instrucciones, Realizar pregunta a Com, Contestar pregunta a Com, Descartar amigos/as Usuario, Ceder turno, Resolver Usuario, Salir, Visualizar detalles amigo/a tablero azul, Visualizar detalles objetivo-Com, Visualizar detalles amigo/a tablero rojo, Visualizar detalles del objetivo-Usuario.
Tipo	Primario
Descripción	Es uno de los actores principales y representa a cualquier persona que desee utilizar la aplicación.

Tabla 1. Actor Usuario.

Actor	Com
Casos de uso	Realizar pregunta a Usuario, Contestar pregunta Usuario, Descartar amigos/as tablero Com, Resolver Com.
Tipo	Primario
Descripción	Es uno de los actores principales. Representa a la IA, de modo que será el oponente al que se enfrentará Usuario.

Tabla 2. Actor COM.



3.4.1.2 Especificación detallada de los casos de uso

En este apartado se realiza la especificación de los casos de uso. Cada caso de uso mostrado en las Tablas 3-19 estará especificado por los siguientes atributos:

- **Identificador:** Identifica al caso de uso de forma única. Debe seguir el formato: CU-XXX, siendo XXX un valor numérico asociado a cada caso de uso.
- **Nombre:** Breve especificación textual del caso de uso.
- **Actores:** Tipo de usuario del sistema que inicia el caso de uso.
- **Objetivo:** Finalidad del caso de uso.
- **Precondiciones:** Estado previo que se debe cumplir para poder realizar la operación asociada al caso de uso.
- **Postcondiciones:** Estado en el que queda el sistema tras realizar la operación asociada.
- **Escenario básico:** Especifica la manera en la que interactúa un actor con el sistema y cuál es la respuesta que el sistema ofrece.
- **Escenarios alternativos:** Condiciones excepcionales que afectan al escenario y respuestas del sistema ante esas situaciones.

Identificador	CU - 001
Nombre	Iniciar juego.
Actores	Usuario.
Objetivo	Iniciar el juego.
Precondiciones	Estar dado de alta en <i>Facebook</i> , conectarse a la red social y a la aplicación.
Postcondiciones	Juego listo para empezar.
Escenario básico	1. Usuario inicia la ejecución de la aplicación. La aplicación hace una consulta a las BB.DD. de <i>Facebook</i> solicitando datos de los amigos/as de Usuario.
Escenarios alternativos	2. Usuario ejecuta <i>iniciar juego</i> . Se realizan el resto de consultas a las BB.DD. de <i>Facebook</i> . 3. La aplicación carga la interfaz mostrando la vista de Usuario (tablero azul).

Tabla 3. Caso de uso CU-001.



Identificador	CU - 002
Nombre	Cambiar vista.
Actores	Usuario
Objetivo	Cambiar la interfaz en la que se encuentra el usuario en ese instante.
Precondiciones	El juego ha de estar iniciado.
Postcondiciones	Se hace visible la vista contraria a la que se encuentra Usuario, ocultando la actual.
Escenario básico	<ol style="list-style-type: none">1. Usuario se encuentra en su vista, tablero azul, y solicita ir a la vista del contrincante (Com).2. La aplicación muestra la vista de Com (tablero rojo) ocultando la de Usuario.
Escenarios alternativos	<ol style="list-style-type: none">1. Usuario se encuentra en la vista de Com, tablero rojo, y solicita ir a su vista.2. La aplicación muestra la vista de Usuario, tablero azul, ocultando la de Com.

Tabla 4. Caso de uso CU-002.

Identificador	CU - 003
Nombre	Ver instrucciones.
Actores	Usuario.
Objetivo	Ver instrucciones y reglas del juego.
Precondiciones	El juego ha de estar iniciado y estar en la vista de Usuario (tablero azul).
Postcondiciones	Instrucciones disponibles para consultar en una página web abierta en otra ventana independiente.
Escenario básico	<ol style="list-style-type: none">1. Usuario encontrándose en su vista (tablero azul) solicita poder observar las instrucciones.2. La aplicación abre las instrucciones en una ventana independiente a aquella en la que se está desarrollando el juego
Escenarios alternativos	

Tabla 5. Caso de uso CU-003.



Identificador	CU - 004
Nombre	Realizar pregunta a Com.
Actores	Usuario
Objetivo	Seleccionar y completar, en caso de que fuera necesario, una pregunta a formular a Com.
Precondiciones	El juego ha de estar iniciado, estar en la vista de Usuario y tiene que ser el turno de éste.
Postcondiciones	Ejecución realizada y se obtiene la respuesta por parte de Com.
Escenario básico	<ol style="list-style-type: none"> 1. Aplicación iniciada, principio de turno de Usuario. 2. Seleccionar pregunta de entre las siguientes: <ol style="list-style-type: none"> 01. ¿Es hombre? 02. ¿Es mujer 03. ¿Su ciudad de nacimiento es ...? 04. ¿Nació en la comunidad de ...? 05. ¿Nació en el país de ...? 06. ¿Su situación sentimental es soltero(a)? 07. ¿Su situación sentimental es casado(a)? 08. ¿Su situación sentimental es "Es complicado"? 09. ¿Su situación sentimental es "Tiene una relación"? 10. ¿Su situación sentimental es prometido(a)? 11. ¿Su situación sentimental es "Tiene una relación abierta"? 12. ¿Su situación sentimental es viudo(a)? 13. ¿Su situación sentimental es separado(a)? 14. ¿Su situación sentimental es divorciado(a)? 15. ¿Nació en el año ...? 16. ¿Nació en el mes ...? 17. ¿Nació en el día ...? 18. ¿Estudió en el centro ...? 19. ¿Reside en la ciudad de ...? 20. ¿Reside en el país de ...? 21. ¿Trabaja en ...? 22. ¿Uno de sus libros favoritos es ...? 23. ¿Una de sus películas favoritas es ...? 24. ¿Uno de sus artistas/grupos musicales favoritos es ...? 25. ¿Uno de sus programas favoritos de tv es ...? 26. ¿Su nombre es ...? 3. Enviar pregunta a la com.
Escenarios alternativos	

Tabla 6. Caso de uso CU-004.



Identificador	CU - 005
Nombre	Contestar pregunta de Com.
Actores	Usuario.
Objetivo	Contestar a pregunta formulada por Com.
Precondiciones	El juego ha de estar iniciado, encontrarse en la vista de Usuario, que éste pueda leer la pregunta formulada por Com y estar en el turno de este último.
Postcondiciones	Com, dependiendo de la respuesta, descartará o no a amigos/as de su tablero.
Escenario básico	<ol style="list-style-type: none">1. Encontrándose en la vista de Usuario, éste puede leer la pregunta formulada por Com.2. Usuario enviará su contestación a la pregunta seleccionando una de las siguientes posibilidades:<ul style="list-style-type: none">• Sí• No• Sin detalles
Escenarios alternativos	

Tabla 7. Caso de uso CU-005.



Identificador	CU - 006
Nombre	Descartar amigos/as tablero usuario.
Actores	Usuario.
Objetivo	Descartar a los amigos/as en función de la respuesta dada por Com a la pregunta previamente formulada.
Precondiciones	El juego ha de estar iniciado, encontrarse en la vista de Usuario, estar en su turno y que éste pueda leer la respuesta dada por Com a la pregunta previamente formulada.
Postcondiciones	Al acabar de descartar, el usuario está en disposición de ceder el turno a Com o seguir descartando.
Escenario básico	<ol style="list-style-type: none">1. Encontrándose en la vista de Usuario, éste puede leer la respuesta de Com a la pregunta previamente formulada.2. Usuario seleccionará a todos aquellos amigos/as del tablero que no cumplan la respuesta.3. Usuario ejecutará descartar, de manera que quedarán descartados todos los amigos/as seleccionados.
Escenarios alternativos	

Tabla 8. Caso de uso CU-006.

Identificador	CU - 007
Nombre	Ceder turno.
Actores	Usuario.
Objetivo	Cambiar de turno, de Usuario a Com.
Precondiciones	El juego ha de estar iniciado, encontrarse en la vista de Usuario, estar en su turno y que haya terminado de descartar.
Postcondiciones	Turno para Com.
Escenario básico	<ol style="list-style-type: none">1. Usuario tiene que haber terminado de descartar.2. Usuario ejecuta turno, cediéndole el turno a Com.
Escenarios alternativos	

Tabla 9. Caso de uso CU-007.



Identificador	CU - 008
Nombre	Resolver Usuario.
Actores	Usuario.
Objetivo	El usuario cree conocer su objetivo y decide resolver para saber si ha ganado o no.
Precondiciones	El juego ha de estar iniciado, encontrarse en la vista de Usuario y estar al principio de su turno.
Postcondiciones	El usuario sabrá si ha ganado la partida, o por el contrario ha de empezar de nuevo.
Escenario básico	<ol style="list-style-type: none">1. Usuario decide resolver antes que formular otra pregunta.2. Usuario selecciona de entre todos los amigos/as del tablero el que cree que es la solución correcta.3. Ejecuta resolver.4. Usuario gana la partida.5. Fin del juego.
Escenarios alternativos	<ol style="list-style-type: none">1. Usuario decide resolver antes que formular otra pregunta.2. Usuario selecciona de entre todos los amigos/as del tablero el que cree que es la solución correcta.3. Ejecuta resolver.4. Usuario no ha seleccionado la respuesta correcta, su tablero se carga otra vez, con nuevos amigos/as y pierde turno.

Tabla 10. Caso de uso CU-008.

Identificador	CU - 009
Nombre	Salir.
Actores	Usuario.
Objetivo	Salir de la aplicación.
Precondiciones	Estar conectado a <i>Facebook</i> y a la aplicación.
Postcondiciones	Finalización de la ejecución de la aplicación.
Escenario básico	<ol style="list-style-type: none">1. Usuario está conectado a la aplicación.2. Usuario ejecuta salir.3. La aplicación finaliza su ejecución.
Escenarios alternativos	

Tabla 11. Caso de uso CU-009.



Identificador	CU - 010
Nombre	Visualizar detalles amigo/a tablero azul.
Actores	Usuario.
Objetivo	Poder observar los detalles de un amigo/a en particular del tablero azul.
Precondiciones	El juego ha de estar iniciado, y encontrarse en la vista de Usuario.
Postcondiciones	Usuario puede contemplar los detalles del amigo/a seleccionado/a del tablero azul.
Escenario básico	1. Usuario ha iniciado la ejecución del juego. 2. Usuario selecciona un amigo/a de su tablero. 3. Usuario ejecuta visualizar detalles de amigo/a. 4. Se hacen visibles los datos de éste.
Escenarios alternativos	

Tabla 12. Caso de uso CU-010.

Identificador	CU - 011
Nombre	Visualizar detalles del objetivo-Com.
Actores	Usuario.
Objetivo	Poder observar los detalles del objetivo de Com.
Precondiciones	El juego ha de estar iniciado, y encontrarse en la vista de Usuario.
Postcondiciones	Usuario puede contemplar los detalles del objetivo de Com.
Escenario básico	1. Usuario ha iniciado la ejecución del juego. 2. Usuario ejecuta visualizar detalles del objetivo de Com desde la vista Usuario. 3. Se hacen visibles los datos de éste.
Escenarios alternativos	

Tabla 13. Caso de uso CU-011.



Identificador	CU - 012
Nombre	Visualizar detalles amigo/a tablero rojo.
Actores	Usuario.
Objetivo	Poder observar los detalles de un amigo/a en particular del tablero rojo.
Precondiciones	El juego ha de estar iniciado, y encontrarse en la vista de Com.
Postcondiciones	Usuario puede contemplar los detalles del amigo/a seleccionado/a del tablero rojo.
Escenario básico	<ol style="list-style-type: none">1. Usuario ha iniciado la ejecución del juego.2. Usuario selecciona un amigo/a del tablero rojo.3. Usuario ejecuta visualizar detalles de amigo/a.4. Se hacen visibles los datos de éste.
Escenarios alternativos	

Tabla 14. Caso de uso CU-012.

Identificador	CU - 013
Nombre	Visualizar detalles del objetivo-Usuario.
Actores	Usuario.
Objetivo	Poder observar los detalles del objetivo de Usuario.
Precondiciones	El juego ha de estar iniciado, y encontrarse en la vista de Com.
Postcondiciones	Usuario puede contemplar los detalles de su propio objetivo.
Escenario básico	<ol style="list-style-type: none">1. Usuario ha iniciado la ejecución del juego.2. Usuario ejecuta visualizar detalles del objetivo de Usuario desde la vista Com.3. Se hacen visibles los datos de éste.
Escenarios alternativos	

Tabla 15. Caso de uso CU-013.



Identificador	CU - 014
Nombre	Realizar pregunta a usuario.
Actores	Com.
Objetivo	Realizar una pregunta a contestar por Usuario.
Precondiciones	El juego ha de estar iniciado, encontrarse en la vista de Usuario, ser el turno de Com y que le quede a éste más de un amigo sin descartar en su tablero rojo.
Postcondiciones	Usuario puede leer la pregunta formulada por Com.
Escenario básico	<ol style="list-style-type: none"> 1. Usuario encontrándose en su vista (tablero azul) solicita el cambio de turno. 2. Com lanza su pregunta a la espera de que Usuario de una respuesta. Elegirá de entre todas estas: <ol style="list-style-type: none"> 01. ¿Es hombre? 02. ¿Es mujer 03. ¿Su ciudad de nacimiento es ...? *03. ¿Tiene especificada su ciudad de nacimiento? 04. ¿Nació en la comunidad de ...? *04. ¿Tiene especificada su comunidad de nacimiento? 05. ¿Nació en el país de ...? *05. ¿Tiene especificado su país de nacimiento? 06. ¿Su situación sentimental es soltero(a)? 07. ¿Su situación sentimental es casado(a)? 08. ¿Su situación sentimental es "Es complicado"? 09. ¿Su situación sentimental es "Tiene una relación"? 10. ¿Su situación sentimental es prometido(a)? 11. ¿Su situación sentimental es "Tiene una relación abierta"? 12. ¿Su situación sentimental es viudo(a)? 13. ¿Su situación sentimental es separado(a)? 14. ¿Su situación sentimental es divorciado(a)? 15. ¿Nació en el año ...? *15. ¿Tiene especificado su año de nacimiento? 16. ¿Nació en el mes ...? *16. ¿Tiene especificado su mes de nacimiento? 17. ¿Nació en el día ...? *17. ¿Tiene especificado su día de nacimiento? 18. ¿Estudió en el centro ...? *18. ¿Tiene especificado dónde estudió o estudia? 19. ¿Reside en la ciudad de ...? *19. ¿Tiene especificada su ciudad de residencia? 20. ¿Reside en el país de ...? *20. ¿Tiene especificado su país de residencia? 21. ¿Trabaja en ...? *21. ¿Tiene especificado dónde trabajó o trabaja? 22. ¿Uno de sus libros favoritos es ...? *22. ¿Tiene especificados sus libros favoritos? 23. ¿Una de sus películas favoritas es ...? *23. ¿Tiene especificadas sus películas favoritas? 24. ¿Uno de sus artistas/grupos musicales favoritos es ...? *24. ¿Tiene especificados grupos o artistas musicales



Escenarios alternativos	<p>favoritos?</p> <p>25. ¿Uno de sus programas favoritos de tv es ...?</p> <p>*25. ¿Tiene especificados sus programas de tv favoritos?</p> <p>26. ¿Su nombre es ...?</p>
--------------------------------	--

Tabla 16. Caso de uso CU-014.

Identificador	CU - 015
Nombre	Contestar pregunta de Usuario.
Actores	Com.
Objetivo	Contestar una pregunta realizada por Usuario.
Precondiciones	El juego ha de estar iniciado, encontrarse en la vista de Usuario, ser el turno de este y debe haber enviado una pregunta a Com.
Postcondiciones	Usuario puede ver la respuesta dada por Com.
Escenario básico	<ol style="list-style-type: none"> 1. Usuario, encontrándose en su vista (tablero azul), envía una pregunta a Com. 2. Com lanza la respuesta de modo que queda visible para Usuario eligiendo una de las siguientes respuestas: <ul style="list-style-type: none"> • Si • No • Sin detalles
Escenarios alternativos	

Tabla 17. Caso de uso CU-015.



Identificador	CU - 016
Nombre	Descartar amigos/as tablero Com.
Actores	Com.
Objetivo	Com descarta todos aquellos amigos/as que no cumplan la respuesta dada por Usuario a la pregunta realizada anteriormente.
Precondiciones	El juego ha de estar iniciado, encontrarse en la vista de Usuario, ser el turno de Com y debe estar esperando la respuesta de Usuario a la pregunta anteriormente realizada por Com.
Postcondiciones	Quedan descartados por Com todos aquellos amigos/as del tablero rojo que no cumplan la respuesta dada por Usuario y le es cedido el turno a este último.
Escenario básico	1. Com recibe la respuesta dada por Usuario y automáticamente descarta a todos aquellos que no cumplan la respuesta de su tablero rojo. 2. Com cede el turno a Usuario.
Escenarios alternativos	

Tabla 18. Caso de uso CU-016.

Identificador	CU - 017
Nombre	Resolver Com.
Actores	Com.
Objetivo	Com decide resolver para ganar la partida.
Precondiciones	El juego ha de estar iniciado, encontrarse en la vista de Usuario, estar al principio del turno de Com y que sólo le quede a éste un amigo por descartar de su tablero rojo.
Postcondiciones	Com gana la partida.
Escenario básico	1. Com comprueba que sólo le queda un amigo/a por descartar. 2. Decide resolver comprobando que ha acertado su objetivo. 3. Com gana la partida. 4. Fin del juego.
Escenarios alternativos	

Tabla 19. Caso de uso CU-017.



3.4.2 Definición de los requisitos del sistema

En este apartado se realiza la extracción de requisitos del sistema con el fin de presentar las principales funcionalidades deseadas por el cliente, sirviendo de base a posteriores fases del ciclo de vida del proyecto. Los requisitos identificados proporcionarán al cliente una visión general de la aplicación de forma completa y sin ambigüedades.

3.4.2.1 Identificación de los requisitos

En esta tarea se realiza la obtención detallada de requisitos de usuario mediante sesiones de trabajo con el cliente. La lista de requisitos obtenida no es definitiva y podrá ser modificada durante el proyecto añadiendo requisitos no contemplados, o bien eliminando o modificando requisitos incorrectos, de forma que se terminen cubriendo todas las funcionalidades expuestas por el cliente.

La definición de requisitos recoge lo que quiere el cliente y lo que necesita, englobando los requisitos obtenidos del usuario en dos grandes categorías:

- **Requisitos de capacidad:** Representan lo que necesitan los usuarios para resolver un problema o lograr un objetivo.
- **Requisitos de restricción:** Son las restricciones impuestas por los usuarios sobre cómo se debe resolver el problema o cómo se debe alcanzar el objetivo.

Cada requisito de usuario debe incluir en su definición una serie de atributos que proporcionen toda la información necesaria para su seguimiento posterior y su clasificación. Estos atributos se describen a continuación:

- **Identificador:** Cada requisito de usuario debe estar identificado de forma única. Este identificador tendrá el siguiente formato: RU-C-nnn o RU-R-nnn, donde:
 - RU: Indica que se trata de un requisito de usuario.
 - C: Indica que se trata de un requisito de usuario de capacidad.
 - R: Indica que se trata de un requisito de usuario de restricción.
 - nnn: Tomará valores numéricos dentro del rango 000-999.
- **Prioridad:** Se asignará una prioridad a cada requisito con el fin de poder realizar una planificación correcta durante fases posteriores. Su clasificación puede tomar los valores: alta, media y baja.
- **Necesidad:** los requisitos clasificados como esenciales para el usuario no pueden ser eliminados, mientras que los demás requisitos estarán siempre sujetos a modificación en el caso de que exista una causa que lo justifique. Por lo tanto, se clasificarán descendientemente de acuerdo a su necesidad de la siguiente forma: esencial, deseable y opcional.



- **Claridad:** Identifica la existencia o carencia de ambigüedad de un requisito, de manera que éste pueda o no ser interpretado de varias formas dependiendo del contexto. Los requisitos serán clasificados de acuerdo a su claridad de forma descendente de la siguiente forma: alta, media y baja.
- **Fuente:** Identifica el origen del requisito, que puede estar en el usuario, en una fuente externa como un documento, o en el propio equipo de desarrollo del proyecto.
- **Estabilidad:** Algunos requisitos pueden no estar sujetos a cambios durante el proyecto debido a su naturaleza, mientras que otros pueden experimentar determinados cambios por el desarrollo de la etapa de diseño o los requisitos de software. Los requisitos poco estables deberán ser observados con mayor cuidado durante el desarrollo del proyecto al ser susceptibles a cambios. Cada requisito se clasificará como estable o como inestable.
- **Verificabilidad:** Indica si el cumplimiento de un requisito en el sistema puede ser susceptible de comprobación, esto es, si se puede verificar que el requisito se ha incorporado en el diseño y que en el sistema se puede verificar su cumplimiento. La verificabilidad de cada requisito se clasificará en: alta, media y baja.



3.4.2.2 Requisitos de capacidad

A continuación se presentan los requisitos de capacidad, los cuales representan lo que necesita el usuario para resolver un problema o lograr un objetivo.

Identificador	RU-C-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Se podrán iniciar partidas desde el sitio web de <i>Facebook</i> .		

Tabla 20. Requisito de usuario RU-C-001.

Identificador	RU-C-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El usuario podrá ver, opcionalmente, el tablero de la IA en cualquier momento del desarrollo del juego para así poder apreciar los avances que éste realiza.		

Tabla 21. Requisito de usuario RU-C-002.



Identificador	RU-C-003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>El usuario podrá ver los detalles de los amigos/as participantes en el juego que forman parte tanto del tablero como de los objetivos de uno y otro participante (Usuario y Com). Estos detalles serán:</p> <p><i>Nombre, sexo, fecha de cumpleaños, lugar de nacimiento y estado sentimental, películas, música, libros, televisión, residencia, historial académico, historial de trabajo, y lugar de nacimiento.</i></p>		

Tabla 22. Requisito de usuario RU-C-003.

Identificador	RU-C-004		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>El usuario podrá consultar un apartado de instrucciones en forma de página web.</p>		

Tabla 23. Requisito de usuario RU-C-004.



Identificador	RU-C-005		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Usuario podrá formular las siguiente preguntas acerca de su objetivo a Com:</p> <ol style="list-style-type: none"> 01. ¿Es hombre? 02. ¿Es mujer 03. ¿Su ciudad de nacimiento es ...? 04. ¿Nació en la comunidad de ...? 05. ¿Nació en el país de ...? 06. ¿Su situación sentimental es soltero(a)? 07. ¿Su situación sentimental es casado(a)? 08. ¿Su situación sentimental es "Es complicado"? 09. ¿Su situación sentimental es "Tiene una relación"? 10. ¿Su situación sentimental es prometido(a)? 11. ¿Su situación sentimental es "Tiene una relación abierta"? 12. ¿Su situación sentimental es viudo(a)? 13. ¿Su situación sentimental es separado(a)? 14. ¿Su situación sentimental es divorciado(a)? 15. ¿Nació en el año ...? 16. ¿Nació en el mes ...? 17. ¿Nació en el día ...? 18. ¿Estudió en el centro ...? 19. ¿Reside en la ciudad de ...? 20. ¿Reside en el país de ...? 21. ¿Trabaja en ...? 22. ¿Uno de sus libros favoritos es ...? 23. ¿Una de sus películas favoritas es ...? 24. ¿Uno de sus artistas/grupos musicales favoritos es ...? 25. ¿Uno de sus programas favoritos de tv es ...? 26. ¿Su nombre es ...? <p>Usuario deberá completar la pregunta en caso de que fuera necesario.</p>		

Tabla 24. Requisito de usuario RU-C-005.



Identificador	RU-C-006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Com podrá realizar las siguientes preguntas acerca de su objetivo a Usuario:</p> <ol style="list-style-type: none"> 01. ¿Es hombre? 02. ¿Es mujer? 03. ¿Su ciudad de nacimiento es ...? *03. ¿Tiene especificada su ciudad de nacimiento? 04. ¿Nació en la comunidad de ...? *04. ¿Tiene especificada su comunidad de nacimiento? 05. ¿Nació en el país de ...? *05. ¿Tiene especificado su país de nacimiento? 06. ¿Su situación sentimental es soltero(a)? 07. ¿Su situación sentimental es casado(a)? 08. ¿Su situación sentimental es "Es complicado"? 09. ¿Su situación sentimental es "Tiene una relación"? 10. ¿Su situación sentimental es prometido(a)? 11. ¿Su situación sentimental es "Tiene una relación abierta"? 12. ¿Su situación sentimental es viudo(a)? 13. ¿Su situación sentimental es separado(a)? 14. ¿Su situación sentimental es divorciado(a)? 15. ¿Nació en el año ...? *15. ¿Tiene especificado su año de nacimiento? 16. ¿Nació en el mes ...? *16. ¿Tiene especificado su mes de nacimiento? 17. ¿Nació en el día ...? *17. ¿Tiene especificado su día de nacimiento? 18. ¿Estudió en el centro ...? *18. ¿Tiene especificado dónde estudió o estudia? 19. ¿Reside en la ciudad de ...? *19. ¿Tiene especificada su ciudad de residencia? 20. ¿Reside en el país de ...? *20. ¿Tiene especificado su país de residencia? 21. ¿Trabaja en ...? *21. ¿Tiene especificado dónde trabajó o trabaja? 22. ¿Uno de sus libros favoritos es ...? *22. ¿Tiene especificados sus libros favoritos? 23. ¿Una de sus películas favoritas es ...? *23. ¿Tiene especificadas sus películas favoritas? 24. ¿Uno de sus artistas/grupos musicales favoritos es ...? *24. ¿Tiene especificados grupos o artistas musicales favoritos? 25. ¿Uno de sus programas favoritos de tv es ...? *25. ¿Tiene especificados sus programas de tv favoritos? 26. ¿Su nombre es ...? 		

Tabla 25. Requisito de usuario RU-C-006.



Identificador	RU-C-007		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Usuario podrá responder a las preguntas formuladas por Com con las siguientes opciones: <ul style="list-style-type: none"> • Si • No • Sin detalles 		

Tabla 26. Requisito de usuario RU-C-007.

Identificador	RU-C-008		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Com responderá a las preguntas formuladas por Usuario con las siguientes opciones: <ul style="list-style-type: none"> • Si • No • Sin detalles 		

Tabla 27. Requisito de usuario RU-C-008.



Identificador	RU-C-009		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Usuario podrá descartar de su tablero a todos aquellos amigos/as que no cumplan la respuesta a la pregunta que realizó con anterioridad a Com.		

Tabla 28. Requisito de usuario RU-C-009.

Identificador	RU-C-010		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Com podrá descartar de su tablero a todos aquellos amigos/as que no cumplan la respuesta a la pregunta que realizó con anterioridad a Usuario.		

Tabla 29. Requisito de usuario RU-C-010.

Identificador	RU-C-011		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Usuario podrá resolver al principio de su turno y cuando considere oportuno durante el desarrollo de la partida seleccionando de su tablero al candidato que cree ser su objetivo a identificar.		

Tabla 30. Requisito de usuario RU-C-011.



Identificador	RU-C-012		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Com podrá resolver, al principio de su turno, cuando en su tablero quede un único amigo/a sin descartar.		

Tabla 31. Requisito de usuario RU-C-012.

Identificador	RU-C-013		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Usuario podrá salir del juego en cualquier momento.		

Tabla 32. Requisito de usuario RU-C-013.

Identificador	RU-C-014		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Usuario podrá ceder el turno cuando haya terminado de descartar.		

Tabla 33. Requisito de usuario RU-C-014.



3.4.2.3 Requisitos de restricción

Son las restricciones impuestas acerca de cómo se debe resolver el problema o cómo se debe alcanzar el objetivo.

Identificador	RU-R-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	El usuario de la aplicación ha de enfrentarse a una máquina con una IA (Com) que le permita desarrollar partidas competidas.		

Tabla 34. Requisito de usuario RU-R-001.

Identificador	RU-R-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Cada participante del juego, Usuario y Com, ha de tener un tablero compuesto por 15 amigos/as, oculto para su adversario, seleccionados al azar de entre todos los que tiene el usuario en su cuenta de <i>Facebook</i> .		

Tabla 35. Requisito de usuario RU-R-002.



Identificador	RU-R-003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Cada participante del juego, Usuario y Máquina, ha de tener un objetivo compuesto por un amigo/a, oculto para su adversario, seleccionado al azar de entre todos los que se encuentran en el tablero de su oponente.		

Tabla 36. Requisito de usuario RU-R-003.

Identificador	RU-R-004		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Disponer de una interfaz gráfica lo más sencilla e intuitiva posible, que propicie una partida fluida.		

Tabla 37. Requisito de usuario RU-R-004.

Identificador	RU-C-005		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La aplicación ha de ser capaz de mostrar al Usuario que un amigo/a de los que forman el tablero ha sido descartado.		

Tabla 38. Requisito de usuario RU-R-005.



Identificador	RU-R-006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La aplicación debe recoger y mostrar todos los detalles personales, y foto de inicio, que aparezcan en <i>Facebook</i> de un amigo/a del Usuario que participe en el juego.		

Tabla 39. Requisito de usuario RU-R-006.

Identificador	RU-R-007		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La aplicación será capaz de determinar cuál de los dos oponentes ha ganado la partida por llegar a descubrir su objetivo mostrando un mensaje por pantalla.		

Tabla 40. Requisito de usuario RU-R-007.

Identificador	RU-R-008		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La aplicación mostrará mensajes de ayuda dando consejos acerca de lo que tiene que hacer el usuario en todo momento.		

Tabla 41. Requisito de usuario RU-R-008.



Identificador	RU-R-009		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La aplicación será capaz de volver a cargar de <i>Facebook</i> todos los amigo/as, tanto el objetivo como del tablero, de Usuario si éste no acertó a la hora de resolver su objetivo.		

Tabla 42. Requisito de usuario RU-R-009.

Identificador	RU-R-010		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Cliente
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Asegurar la correcta visualización de la aplicación en los navegadores Microsoft Internet Explorer®, Mozilla Firefox® y Google Chrome®.		

Tabla 43. Requisito de usuario RU-R-010.



3.5 Estudio de alternativas de solución

Con la realización de este proyecto se pretende desarrollar una *Aplicación Rica en Internet (RIA)* que aproveche la conectividad que ofrece *Facebook* a los desarrolladores con sus bases de datos y emplear estos datos durante el desarrollo del juego. La aplicación a desarrollar está basada en el juego de mesa “¿Quién es quién?”. La solución ha sido impuesta por el cliente y no aplica realizar un estudio de diferentes alternativas.

Para desarrollar el producto se utilizará la tecnología *Flex* que une el lenguaje *MXML* y *ActionScript*, utilizando para ello la herramienta creada por *Adobe*, *Flash Builder 4*.

El servidor escogido para alojar nuestra aplicación se trata de uno gratuito y fiable, de modo que se produce un ahorro de costes notable en este aspecto.

4. GESTIÓN DEL PROYECTO

La gestión de proyectos tiene como finalidad principal la planificación, el seguimiento y control de las actividades y de los recursos humanos y materiales que intervienen en el desarrollo.

4.1 Ciclo de vida

El término ciclo de vida del software describe el desarrollo de software, desde la fase inicial hasta la fase final. El propósito de este programa es definir las distintas fases intermedias que se requieren para validar el desarrollo de la aplicación, es decir, para garantizar que el software cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo: se asegura de que los métodos utilizados son apropiados. (Cruz, 2010).

Para el presente proyecto el ciclo de vida elegido ha sido el modelo por prototipos, que es el que más se adapta a las necesidades de desarrollo en este caso, ya que el propio producto a lo largo de su desarrollo puede considerarse una sucesión de prototipos que progresan hasta llegar al estado deseado.

Si no se conoce exactamente cómo desarrollar un determinado producto o cuáles son las especificaciones de forma precisa suele recurrirse a definir especificaciones iniciales para hacer un prototipo, o sea, un producto parcial y provisional. En este modelo, figura 42, el objetivo es lograr un producto intermedio, antes de realizar el producto final, para poder conocer mediante éste cómo responderán las funcionalidades previstas para el producto final.

Debido al uso de *Adobe Flash Builder 4* para desarrollar este proyecto, y a su consiguiente desconocimiento previo de su comportamiento por parte del equipo de desarrollo, la incertidumbre sobre los resultados a obtener era alta. Con la elección de este tipo de ciclo de vida se permite al equipo de desarrollo poder ir realizando prototipos en los que evaluar los resultados pudiéndose marcar nuevos requerimientos para el siguiente prototipo.

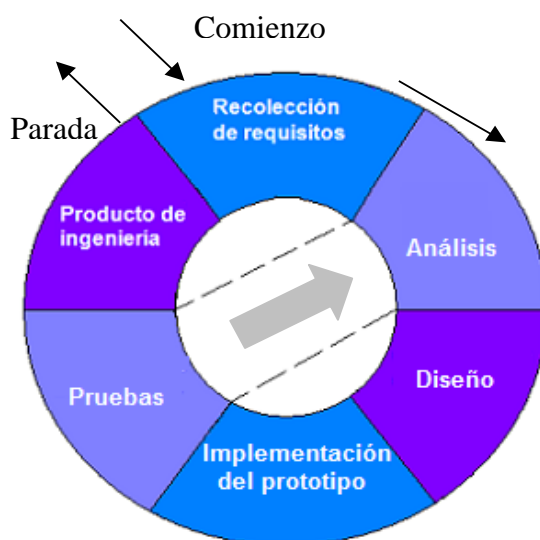


Figura 42. Ciclo de vida. Modelo de prototipo



Este ciclo de vida ha sido considerado como el más adecuado debido a los siguientes motivos:

- Permite evaluar distintas posibilidades de desarrollo, ya que en los prototipos se pueden añadir o eliminar funcionalidades.
- Desconocimiento de la tecnología, pudiendo mejorar aspectos técnicos en cada iteración debido a que se amplían conocimientos y experiencias con dicha tecnología.
- El prototipo actúa como un avance de lo que puede ser el producto final.
- Se experimenta sobre los aspectos del sistema que representan mayor complejidad.
- El usuario reacciona mucho mejor ante un prototipo, sobre el que puede experimentar, que no sobre una especificación escrita.
- Se incrementa la calidad del producto final, ya que el prototipo permite trabajar sobre él.

A continuación se describe cada una de las fases de las que está compuesto el ciclo de vida elegido:

- **Recolección de requisitos:** Se realiza una primera toma de los requisitos software del sistema con el cliente obteniendo un primer boceto sobre lo que debería ser la aplicación a desarrollar.
- **Análisis:** El equipo de desarrollo realiza un análisis del prototipo, se determinan (o amplían) los requisitos software del sistema, especificando a un alto nivel la arquitectura de la solución que se propone para dichos requisitos.
- **Diseño:** Se realiza un diseño preliminar del software que posteriormente se irá ampliando en cada una de las iteraciones del ciclo de vida. Se diseña la interfaz de usuario y la arquitectura a un nivel más bajo detallando los procesos del sistema.
- **Implementación del prototipo:** En esta etapa se codifican los interfaces de usuario, los procesos y se documenta el manual de usuario.
- **Pruebas:** Se realizan una serie de pruebas destacando los puntos débiles, o a cambiar, para poder ser modificados en posteriores fases del ciclo de vida, o por el contrario estas pruebas pueden superarse satisfactoriamente predeterminando el final de la iteración y pasar a la última etapa.
- **Producto de ingeniería:** Se realizan los últimos retoques, así como la documentación necesaria para poder obtener un producto totalmente finalizado para ser entregado al cliente.

4.2 Organización del proyecto

A la hora de realizar cualquier proyecto se hace necesaria la tarea de planificación detallando el tiempo necesario estimado para cumplir las distintas actividades y tareas que forman parte del proyecto así como los recursos asignados a cada una de ellas. Por ello, se ha desarrollado una planificación de trabajo en función del ciclo de vida seleccionado anteriormente.

El RBS (*Resource Breakdown Structure*) es una representación jerárquica de los recursos, tanto humanos como materiales, necesarios para la planificación de un proyecto. (Wikipedia, 2011). Esta técnica de organización de proyectos software tiene por objeto representar la organización humana del proyecto, así como la estructura de recursos tecnológicos y materiales.

En lo que se refiere a los recursos humanos que intervienen en el desarrollo del proyecto, a continuación se muestran las personas y roles que participan en él con sus funciones:

- **Jefe de Proyecto:** Se encarga de la gestión, organización, planificación y supervisión del proyecto a lo largo de todo su desarrollo.
- **Analista:** Se encarga de obtener y redactar los requisitos de usuario además de modelar los procesos y tareas a codificar.
- **Diseñador:** Su tarea es el diseño de las interfaces, la arquitectura del sistema y el plan de verificación y validación.
- **Programador:** Se encarga de la codificación del sistema así como de llevar a cabo las pruebas necesarias sobre el mismo.

Los recursos materiales vienen definidos en el apartado 5.2.3.3 *Entorno operacional*. El diagrama mostrado en la figura 43 muestra el RBS del presente proyecto.

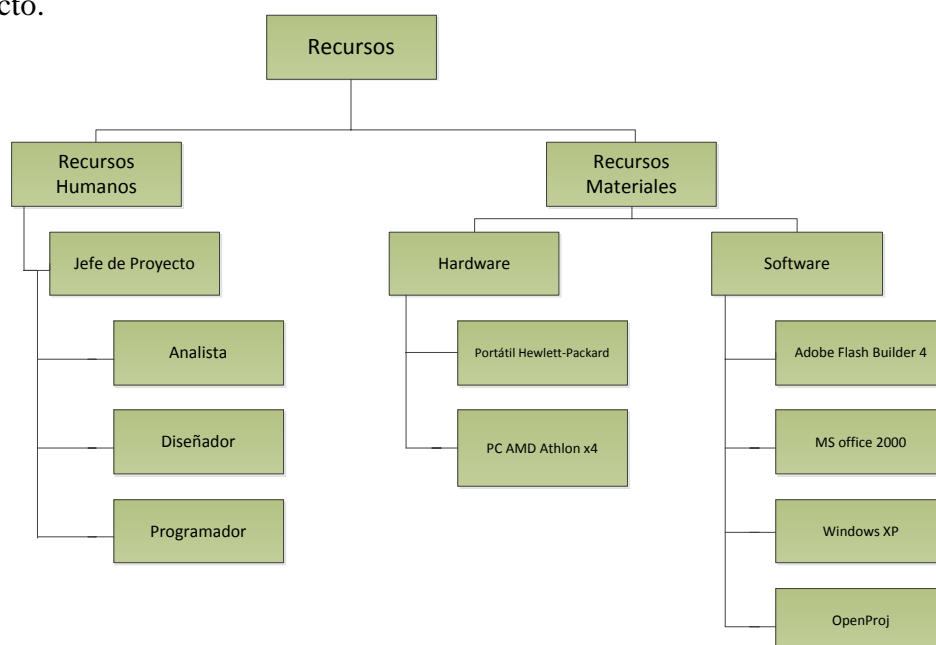


Figura 43. RBS.

El WBS (*World Breakdown Structure*) es a herramienta simple y práctica que permite definir el alcance de un proyecto. (Wikipedia, 2011). Es la base para definir qué se debe hacer y realizar la planificación del proyecto, identificando y definiendo las tareas que se van a realizar. Además, el WBS permite observar la descomposición de los distintos procesos en subprocesos y éstos, a su vez, en tareas agrupadas según funcionalidad. La base en la que está basada el WBS es el modelo de proceso software IEEE 1074. La aplicación de esta herramienta en el proyecto se puede ver en las figuras 34 - 39.

En el primer proceso se realiza el estudio de viabilidad del proyecto, figura 44. En éste se analizará la solicitud del cliente, la situación actual y si el desarrollo del proyecto es viable. Además se recogerá dicha información en el capítulo de Estudio de Viabilidad.

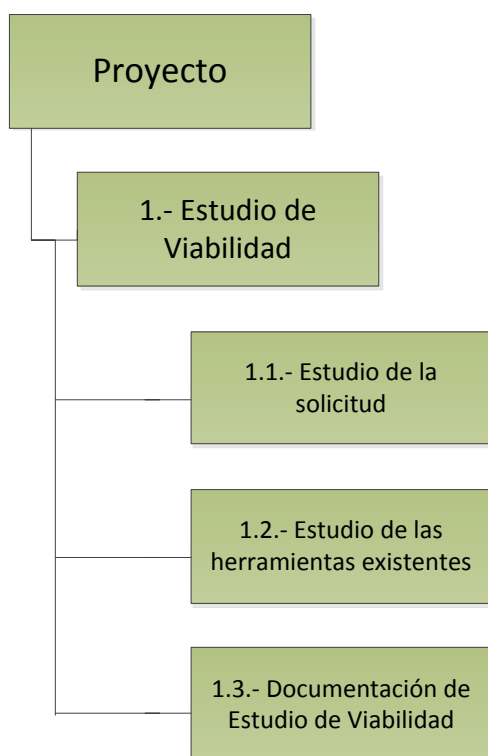


Figura 44. Proceso WBS. Estudio de Viabilidad

Una vez aceptado el desarrollo del proyecto mediante el estudio de viabilidad, el siguiente paso a realizar es la gestión del proyecto, figura 45. En este proceso se debe seleccionar el modelo de ciclo de vida que más se adapte al proyecto y se realizará tanto la planificación a seguir durante el desarrollo del proyecto como una estimación del coste de éste.

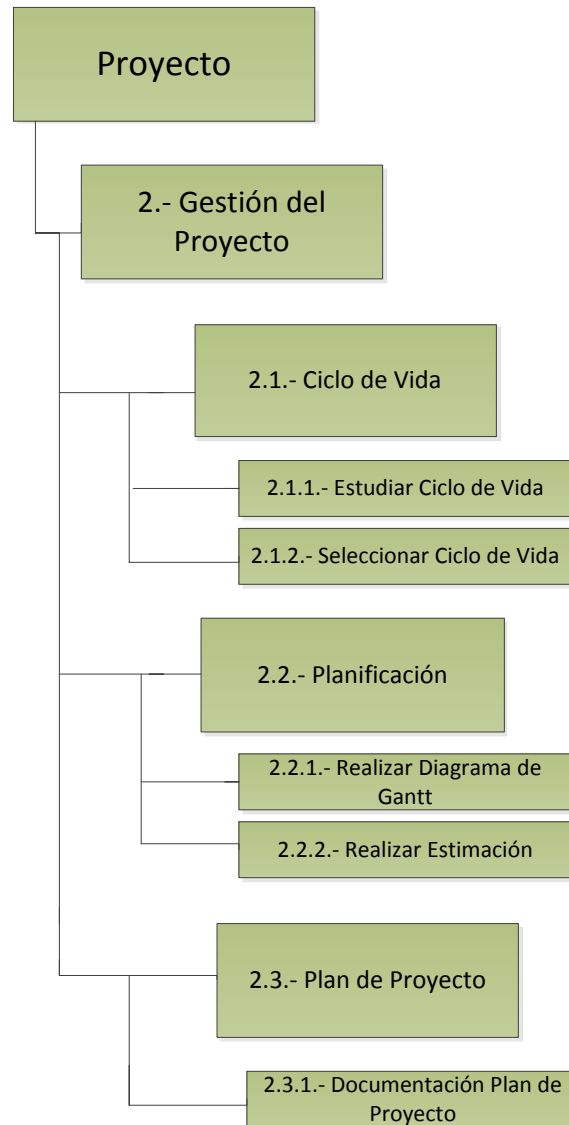


Figura 45. Proceso WBS. Gestión del Proyecto

A continuación se comenzará con las iteraciones realizadas en el modelo de ciclo de vida elegido para el desarrollo del sistema (en total tres). En cada iteración se deben llevar a cabo las tareas propias del análisis como son la recopilación de especificaciones y requisitos, o la selección de las herramientas de desarrollo más apropiadas, figura 46. La realización de estas tareas es imprescindible, ya que son las que permiten obtener el diseño más eficiente del software en la fase posterior. Después del análisis vendrá la fase de diseño. En ésta se van a desarrollar las tareas relacionadas con la arquitectura del sistema y la plataforma del proyecto. La siguiente fase será la de implementación, que



viene representada mediante el subproceso del mismo nombre, y que consistirá, principalmente, en la codificación de la interfaz y de los procesos, ambos diseñados con anterioridad. Por último, se hace imprescindible el subproceso de pruebas, en el que se definirán y ejecutarán las pruebas para la verificación y validación del prototipo.

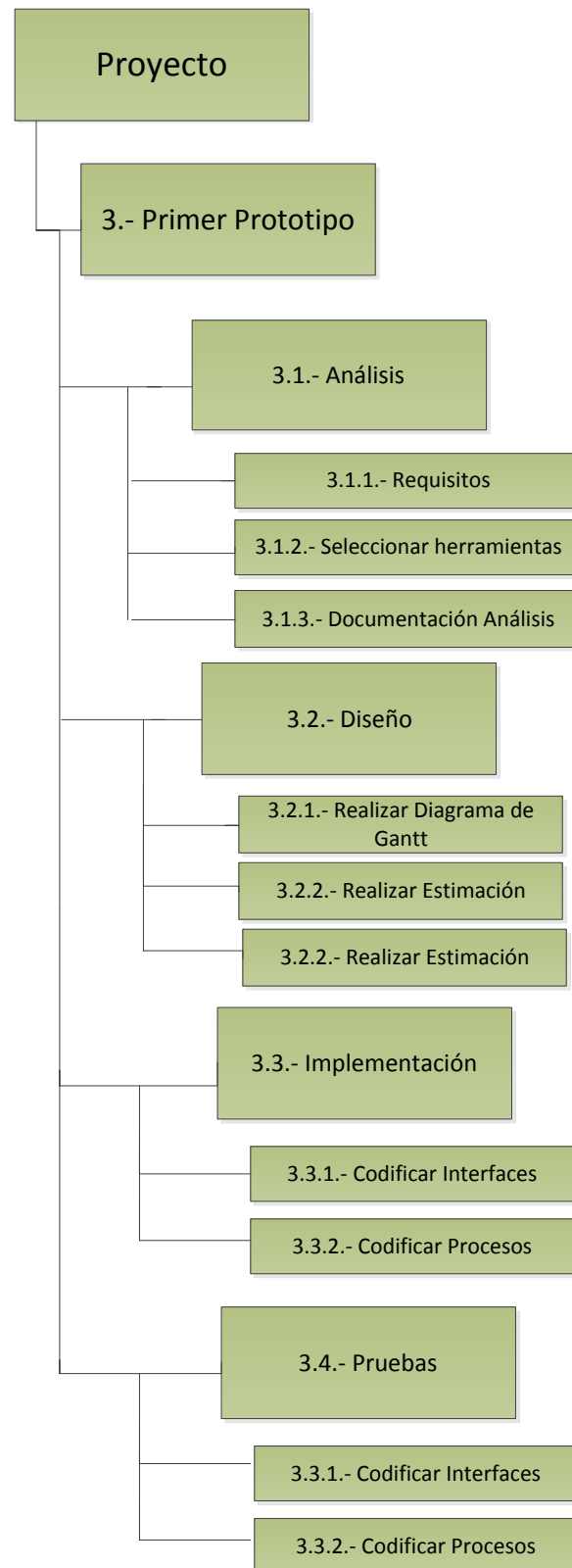


Figura 46. Proceso WBS. Primer Prototipo



En el primer prototipo, figura 46, se realizará una versión escritorio de la aplicación, es decir, no hará falta alojar la aplicación en ninguna web porque ésta se ejecutará en un ordenador propio. De esta manera se simplifica la tarea de pruebas de esta iteración.

En el segundo prototipo, figura 47, se hará una versión web de la aplicación. Por este motivo ésta ya debe ser alojada en un servidor para poder ser probada.

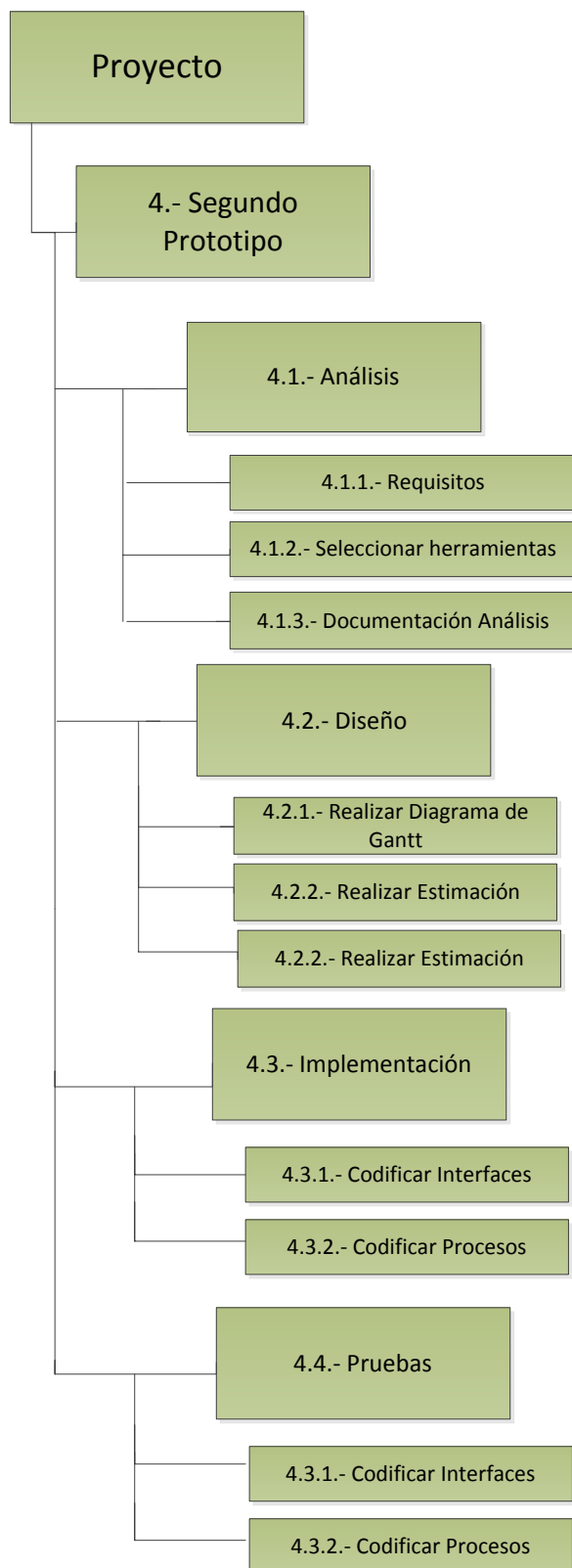


Figura 47. Proceso WBS. Segundo Prototipo



En la última iteración, figura 48, se realizarán los cambios oportunos, sobre todo de interfaz, para que la aplicación web pueda ser ejecutada bajo la página web de *Facebook*.



Figura 48. Proceso WBS. Tercer Prototipo



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

Una vez que el último prototipo desarrollado sea aprobado por el cliente, o tutor, se pasará a un último proceso, figura 49, en el que se realizará el manual de usuario para que éste pueda utilizar todas las funcionalidades disponibles en el sistema con mayor facilidad.

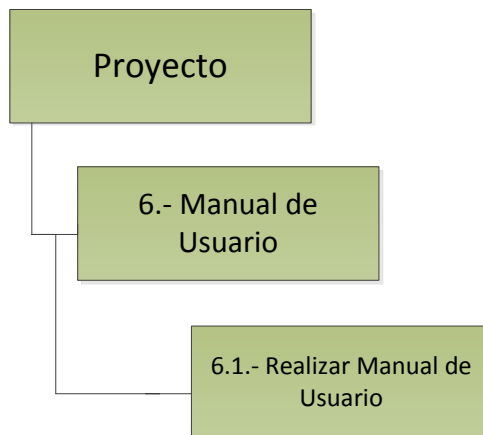


Figura 49. Proceso WBS. Manual de Usuario

4.3 Planificación

La duración total del proyecto es de 181 días laborales, comenzando el desarrollo el día 7 de Febrero de 2011 y terminando el 15 de Noviembre de 2011, cuando el sistema se encuentra finalizado y documentado para su entrega al cliente , figura 50.



Figura 50. Diagrama de Gantt. General.

El proyecto comienza con el estudio de viabilidad, figura 51. El jefe de proyecto será el encargado de esta tarea realizando un estudio tanto de la solicitud del cliente como de la situación actual, posteriormente elaborando un documento (capítulo) que recoja toda la información generada durante este proceso. Por último, este documento mostrará si la solicitud realizada es factible de ser llevada a cabo o no.



Figura 51. Diagrama de Gantt. Estudio de Viabilidad

Una vez superada esta primera fase, y considerando viable el proyecto, comienza la fase de gestión de proyecto, figura 52. Esta fase también será llevada a cabo por el Jefe del Proyecto. En ella se realizará un estudio de los ciclos de vida existentes y se seleccionará el que se considere más apropiado para el desarrollo del proyecto, documentando las razones y el motivo de la elección.

A continuación, el jefe de proyecto realizará el diagrama de Gantt en el cuál figuran las tareas y actividades necesarias para desarrollar el producto así como la duración estimada y los recursos necesarios de cada una de ellas. En el diagrama de la figura 40 las tareas aparecen como finalizadas puesto que ya se ha concluido el proyecto.

Para finalizar esta fase, se ha de realizar un documento de Gestión de Proyecto (que se corresponderá con el capítulo con el mismo nombre en el presente trabajo), en el que se incluya toda la planificación y estimaciones realizadas. Una vez hecho esto se dará



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

comienzo al seguimiento del proyecto que se llevará a cabo en paralelo a todas las actividades a realizar hasta la entrega final de la aplicación.

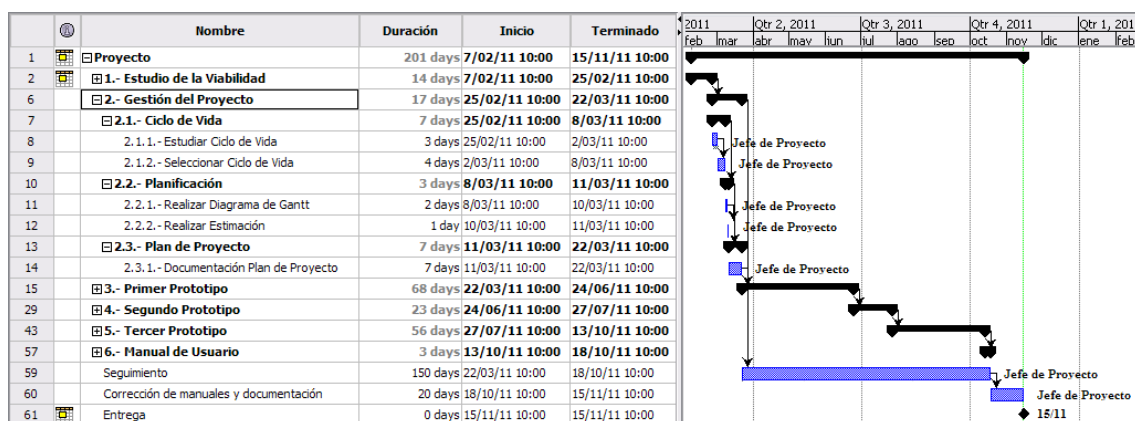


Figura 52. Diagrama de Gantt. Gestión del Proyecto

Una vez finalizadas todas las tareas de la fase de Gestión de Proyecto se iniciará el desarrollo del primer prototipo, figura 53. Este desarrollo seguirá los siguientes pasos:

1. El analista deberá recopilar los requisitos del usuario para obtener los requisitos software y seleccionar así las herramientas necesarias para satisfacerlos.
2. El diseñador leerá toda la documentación generada por el analista y realizará el diseño de la arquitectura del sistema, definiendo tanto las interfaces como los procesos a implementar.
3. El programador implementará el sistema según el diseño realizado para poder así alcanzar todos los requisitos del usuario.
4. Realización de una batería de pruebas que servirán para verificar el correcto funcionamiento del prototipo y la evaluación por parte del cliente o tutor. Es responsabilidad del cliente trabajar con el prototipo y evaluar sus características y operación.

Como se ve en la figura 53, en este prototipo los mayores esfuerzos van a parar a las tareas de recolección de requisitos y codificación, sobre todo de los procesos, ya que será en este primer prototipo donde se implementen la mayoría de funcionalidades de la aplicación.

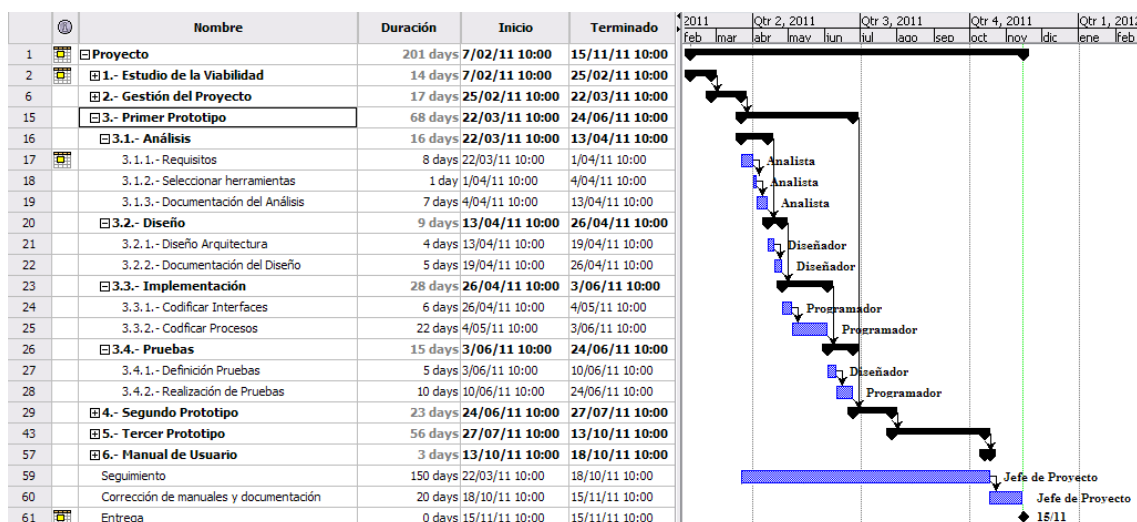


Figura 53. Diagrama de Gantt. Primer prototipo

El analista comienza la segunda iteración partiendo de la información aportada por el usuario en la evaluación del prototipo anterior, figura 54. Se volverán a planificar los cambios a realizar con el cliente, y será otra vez el analista el encargado de realizar dicha tarea.

Al igual que en el primer prototipo: el analista documentará los nuevos requisitos y seleccionará las herramientas necesarias; el diseñador leerá la documentación generada en el paso anterior y diseñará las nuevas interfaces y funcionalidades; el programador codificará todo; y, finalmente, se realizarán las pruebas para verificar que todo funciona como se esperaba.

Este prototipo básicamente trata de convertir la aplicación anterior en una aplicación web, por lo que apenas hay que modificar procesos e interfaces y los mayores esfuerzos se centran en el diseño de la arquitectura.

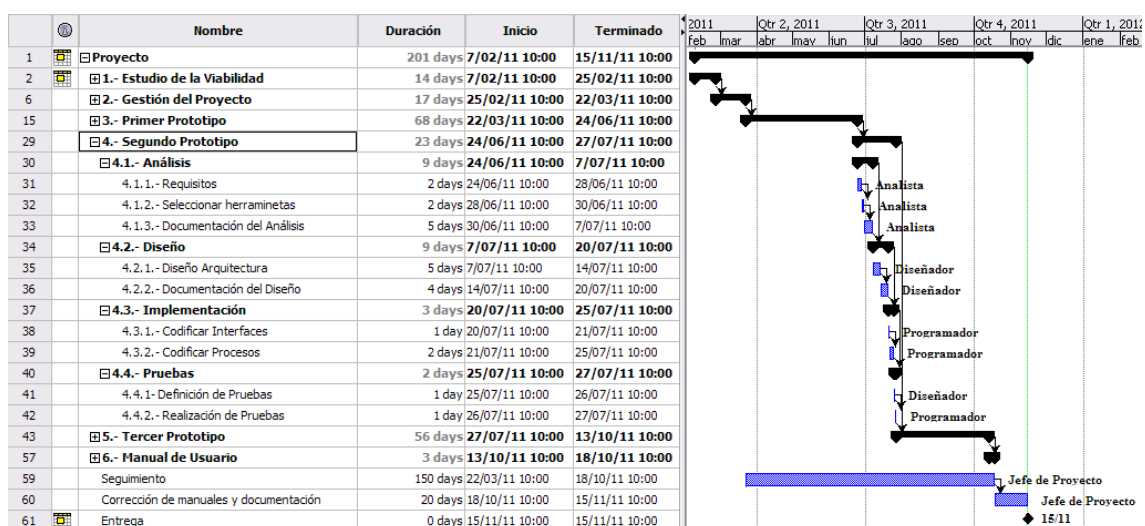


Figura 54. Diagrama de Gantt. Segundo prototipo.

A continuación se desarrollará el tercer prototipo, figura 55. En éste se intentarán satisfacer todos los requisitos expuestos por el cliente a lo largo del proyecto. El proceso a seguir será igual al de los dos anteriores prototipos.

En este caso los esfuerzos se centrarán en la realización de una interfaz que se integre lo máximo posible en el *iFrame* de *Facebook*, y se realizará una batería de pruebas exhaustiva para dar el producto por finalizado.

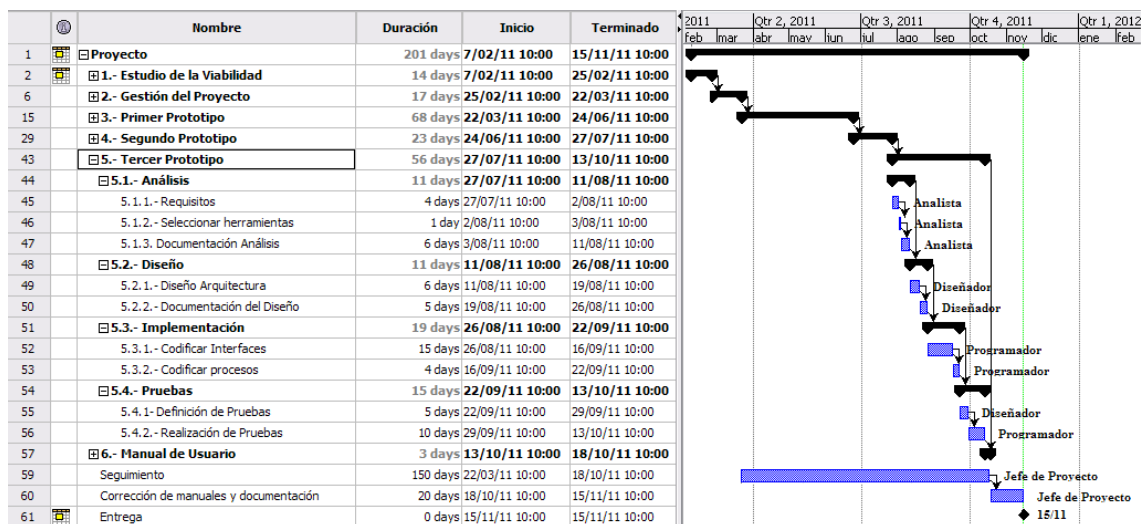


Figura 55. Diagrama de Gantt. Tercer prototipo.

usuario, figura 56, indicando los aspectos de la aplicación así como sus funciones y como llevar a cabo una partida con total fluidez.



Figura 56. Diagrama de Gantt. Manual de usuario.

Finalmente se realizará una corrección de manuales y documentación, figura 57. En esta tarea, el tutor del proyecto revisará toda la documentación generada durante el desarrollo del software señalando una serie de cambios a realizar. Posteriormente, el jefe de proyecto realizará las correcciones pertinentes obteniendo así la documentación final para poder ser entregada a los clientes.

El último hito del proyecto es la entrega de la documentación y el software desarrollado al cliente o tutor. Tras esto se da por concluido el proyecto el 18 de Octubre de 2011.



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook



Figura 57. Diagrama de Gantt. Corrección de manuales y entrega.

4.4 Estimación de costes

Para calcular el coste estimado del proyecto se han tenido en cuenta los costes tanto materiales como humanos que podía generarse a lo largo del desarrollo del proyecto.

En primer lugar se calcula el coste asociado a las herramientas utilizadas para el desarrollo del proyecto. Se decidió realizar el mayor ahorro en este apartado, por lo que, algunas de las herramientas utilizadas corresponden a software libre o se han conseguido licencias especiales para estudiantes, como por ejemplo *Adobe Flash Builder 4*, el cuál *Adobe* cedió la licencia. Con el fin de poder comprobar el dinero ahorrado se muestra una comparativa en la tabla 44.

Herramienta	Coste licencia	Coste para el proyecto
Microsoft Windows Xp	114,00 €	114,00 €
Microsoft Office 2010	379,00 €	379,00 €
Adobe Flash Builder 4	216,00 €	Gratuito
Microsoft Project 2010	436,00 €	OpenProj + StarUml
Total (IVA INCLUIDO)	1145,00 €	493,00 €

Tabla 44. Costes de licencias.

Como se puede observar en dicha tabla, se ha producido un ahorro de 652 € provenientes de:

- **Adobe Flash Builder 4:** ahorro de 216,00 € gracias a la cesión de la licencia por parte de la propia *Adobe* a estudiantes de ingeniería.
- **Microsoft Project 2010:** ahorro de 436,00 €, gracias al uso de las herramientas de software libre *OpenProj* y *StarUML*, ambas de gran potencia.

Finalmente, el gasto derivado de licencias y software en general asciende a 493,00€.

Para el cálculo de los costes asociados a los recursos humanos se ha tomado una jornada laboral de 8 horas, distribuidas en horarios de 9:00 a 15:00 y de 16:00 a 18:00. El proyecto ha sido generado por un solo ingeniero, el cual se ha adaptado a los distintos roles anteriormente descritos. La tabla 45 muestra una relación entre las funciones que se han adoptado, el tiempo invertido y su coste.



Recurso	Horas	Precio/Hora	Total
Jefe de proyecto	336	20 €	6.720 €
Analista	288	15 €	4.320 €
Diseñador	240	15 €	3.600 €
Programador	592	10 €	5.920 €
Total			20.560 €

Tabla 45. Costes de recursos humanos

Analizando dicha tabla se puede observar que el coste total resultante de las horas dedicadas por el desarrollador asciende a 20.560 €.

Por lo tanto, la suma de los costes materiales y humanos hace que el costo total del desarrollo del proyecto sea el indicado en la tabla 45.

Recursos materiales	493,00 €
Recursos humanos	20.560,00 €
Total	21.053 €

Tabla 46. Coste total del proyecto.



5. ANÁLISIS DEL SISTEMA

En esta parte del documento se realiza una especificación detallada del sistema que satisfaga las necesidades de información de los usuarios y sirva de base para el posterior diseño del sistema.

5.1 Introducción

Esta etapa consiste principalmente en darle relevo a la información de la que se dispone actualmente por otra más específica y proponer los rasgos generales de la solución futura.

5.1.1 Objetivo del Análisis del Sistema

Al realizar el Análisis del Sistema se pretende obtener una colección completa y detallada de los requisitos del sistema tomando como punto de partida los requisitos identificados en el Estudio de Viabilidad del Sistema.

La información generada en esta fase será la base para el desarrollo de la fase de Diseño del Sistema, en el que se especificará el diseño completo del sistema de información.

5.1.2 Alcance

Esta fase tiene como objetivo obtener una especificación detallada del sistema que se va a diseñar. Mediante su producto, el capítulo de Análisis del Sistema, se pretende captar cuales son las necesidades que tiene el cliente.

En primer lugar, se definirá cual es el alcance del sistema que se desea desarrollar, así como el entorno tecnológico que se va a asociar al proyecto. Además, se identificarán a todos los participantes y usuario finales que aparecen a lo largo de la vida del proyecto.

A continuación, se definirán los requisitos software que debe cumplir el sistema a desarrollar tomando como punto de partida los casos de uso y los requisitos de usuario especificados en la anterior fase (Estudio de Viabilidad del Sistema).

Finalmente, serán definidos las interfaces de usuario que van a ser utilizadas para terminar con las comprobaciones de calidad sobre los distintos modelos y requisitos software que se generaron durante la fase de análisis.



5.2 Definición del Sistema

En este apartado se determina el alcance del sistema a desarrollar, para que permita satisfacer las necesidades planteadas por el cliente y recogidas en el capítulo de Estudio de Viabilidad del Sistema. Se identificarán los estándares y normas que se emplearán, las restricciones generales y el entorno de desarrollo, así como los usuarios implicados en dicho desarrollo.

5.2.1 Determinación del alcance del sistema

El sistema a desarrollar consiste en una aplicación para *Facebook* que permita a un usuario de dicha red social participar en partidas, contra una IA, basadas en el popular juego de mesa “¿Quién es quién?”.

Las características fundamentales a destacar del desarrollo del producto son:

- La aplicación debe estar alojada en un servidor gratuito de internet.
- La aplicación debe funcionar bajo la web de *Facebook*
- Se desarrollará una IA que permita a los usuarios de la aplicación participar en partidas competidas.
- La aplicación accederá a las bases de datos que pone *Facebook* a disposición de los desarrolladores para utilizar dichos datos durante las partidas.
- El software tendrá una interfaz intuitiva y atractiva para los usuarios.

5.2.2 Identificación del entorno tecnológico

Para el desarrollo del software, el entorno tecnológico que se va a emplear no es muy exigente, tanto software como hardware. El entorno de desarrollo que se va a emplear es el *Adobe Flash Builder 4* y este no requiere de equipos con alta potencia. Véase el apartado 5.2.3.3 *Entorno operacional* para observar en mayor profundidad todos los elementos empleados para el desarrollo.

5.2.3 Especificación de estándares y normas

Los estándares y normas que ha de respetar el proceso de desarrollo del presente proyecto son:

- **Métrica Versión 3:** Los capítulos de Estudio de Viabilidad del Sistema, Análisis del Sistema y Diseño del Sistema generados durante el proyecto seguirán una adaptación de la metodología de desarrollo *Métrica Versión 3*. La metodología será adaptada a las necesidades del proyecto, de manera que determinadas tareas no se realizarán o no se profundizará demasiado en ellas.



- **IEEE 1074:** Es la norma utilizada en la Gestión del Proyecto para la definición del modelo de procesos planificado para el desarrollo del proyecto.
- **UML:** Es el lenguaje de modelado de sistemas que se utilizará para realizar el diseño del sistema.

5.2.3.1 Restricciones generales

A continuación se detallan las restricciones que deberá cumplir el sistema a diseñar:

- En primer lugar, el sistema funcionará en todos los equipos que tengan instalado *Adobe Flash Player* en su versión 10.2 (como mínimo).
- La interfaz con la que interaccionarán los usuarios finales debe proporcionar todas las funcionalidades descritas en los requisitos software del sistema de una forma intuitiva y atractiva, evitando posibles ambigüedades que puedan ocasionar confusión al usuario.
- El idioma usado en el desarrollo del sistema será el español, aunque durante la implementación será utilizado el inglés para nombrar a funciones, procedimientos, variables, etc. Por el contrario, la interfaz de la aplicación estará totalmente en castellano.

5.2.3.2 Supuesto y dependencias

Todos los usuarios participantes en la presente fase de análisis conocen el estándar empleado (*Métrica Versión 3*), por lo que todos los productos generados durante el proceso deben de respetar lo establecido por dicho estándar.

5.2.3.3 Entorno operacional

El sistema se desarrollará mediante el entorno de desarrollo *Adobe Flash Builder 3*, utilizando como lenguaje de programación *Flex* que combina *ActionScript* y *MXML*. Los usuarios del sistema tendrán que disponer de un ordenador en el cual el sistema operativo debe tener instalado *Adobe Flash Player* en su versión 10.2 (como mínimo).

Para el desarrollo del proyecto se han utilizado dos ordenadores con las siguientes características.

- **Ordenador Personal:**
 - Microprocesador: AMD Athlon x4.
 - Memoria Ram: 4096 MB.
 - Sistema operativo: Microsoft Windows XP.
- **Ordenador portátil Hewlett-Packard:**



- Microprocesador: Intel Core 2.
- Memoria Ram: 1024 MB.
- Sistema operativo: Microsoft Windows XP.

En ambos ordenadores se contaba con la *suite* ofimática *Microsoft Office 2010* para la gestión de documentos.

5.2.3.4 Identificación de los usuarios

En este apartado se van a detallar los usuarios que participan en el proceso de análisis del sistema así como los usuarios que lo validarán y aceptarán finalmente.

Con respecto a los participantes en el proceso de análisis:

- **Equipo de desarrollo:** Grupo de personas encargado de llevar a cabo el desarrollo del presente proyecto.
- **Cliente o tutor:** Espera un correcto desarrollo del producto final en términos de calidad, tiempo y coste. Además, es parte importante a la hora de la obtención de los requisitos del sistema a diseñar, ya que el objetivo del proyecto es obtener un producto que cumpla con las necesidades que le han llevado a solicitar el sistema.

En lo que a los **usuarios** finales del sistema se refiere, serán aquellas personas que, con cuenta en *Facebook*, harán uso del sistema desarrollado.

5.2.3.5 Estudio de la seguridad requerida en el proceso de análisis

El acceso a los productos generados en la presenta fase de análisis estará limitado a los usuarios participantes que se describen en el apartado anterior (5.2.3.4 *Identificación de los usuarios*).



5.3 Establecimiento de Requisitos de Software

En este apartado se completará el catálogo de requisitos de modo que el producto final generado pueda ajustarse a estos requisitos.

5.3.1 Obtención de requisitos

A continuación se representarán los Requisitos de Software identificados en el análisis. Estos requisitos son una evolución de los requisitos de usuario definidos en el Estudio de Viabilidad del Sistema, a la vez que otros requisitos añadidos por el equipo de desarrollo.

Los tipos de Requisitos de Software identificados en el proceso han sido los siguientes:

- **Requisitos funcionales (F):** son aquellos requisitos que especifican el propósito del sistema y que derivan directamente de los casos de uso.
- **Requisitos de interfaz (I):** especifican el hardware y/o software con los que debe interactuar el sistema.
- **Requisitos de operación (O):** son todos aquellos requisitos que especifican como se va a resolver el problema.
- **Requisitos de documentación (D):** requisitos que especifican criterios de realización de la documentación asociada al proyecto.
- **Requisitos de calidad (C):** estos requisitos indican tareas o criterios que guiarán a los implicados en el desarrollo del software pudiendo alcanzar las cotas deseadas de calidad.

Cada requisito software debe incluir una serie de atributos en su definición que proporcionarán toda la información necesaria para su posterior seguimiento y clasificación. Estos atributos son:

- **Identificador:** Cada requisito debe estar identificado de forma única. Este identificador tendrá el formato *RS-XX-*nnn**, donde:
 - **RS:** Indica que se trata de un Requisito Software.
 - **XX:** Indicará el tipo de requisito software al que corresponde (ver listado anterior).
 - **nnn:** Tomará valores numéricos entre 000-999, enumerando los requisitos.
- **Prioridad:** Se asignará una prioridad a cada requisito con el fin de poder realizar una planificación correcta durante fases posteriores. Su clasificación puede tomar los valores: alta, media y baja.



- **Necesidad:** Los requisitos clasificados como esenciales no pueden ser eliminados, mientras que los demás requisitos estarán siempre sujetos a modificación en el caso de que exista una causa que lo justifique. Por lo tanto, se clasificarán descendientemente de acuerdo a su necesidad de la siguiente forma: esencial, deseable y opcional.
- **Claridad:** Identifica la falta o existencia de ambigüedad de un requisito, esto es, si puede ser interpretado de varias formas dependiendo del contexto. Los requisitos serán clasificados, de acuerdo a su claridad de forma descendente de la siguiente forma: alta, media y baja.
- **Fuente:** Identifica el origen del requisito, que puede estar en un requisito de usuario o provenir del propio equipo de desarrollo.
- **Estabilidad:** Algunos requisitos pueden no estar sujetos a cambios durante el proyecto debido a su naturaleza mientras que otros pueden estar sujetos a determinados cambios por el desarrollo de la etapa de diseño. Los requisitos poco estables deberán ser observados con mayor cuidado durante el desarrollo del proyecto al ser susceptibles a cambios. Cada requisito se clasificará en estable o inestable.
- **Verificabilidad:** Indica si el cumplimiento de un requisito en el sistema puede ser susceptible de comprobación, esto es, si se puede verificar que el requisito se ha incorporado en el diseño y que en el sistema se puede verificar su cumplimiento. La verificabilidad de cada requisito se clasificará en alta, media y baja.



5.3.1.1 Requisitos Funcionales

Identificador	RS-F-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-001
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Se podrá acceder a la aplicación desde el sitio web de <i>Facebook</i> . Para conectarse a ésta <i>Facebook</i> requerirá el nombre de usuario y contraseña.		

Tabla 47. Requisito software RS-F-001.

Identificador	RS-F-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-001
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Se podrá acceder a la aplicación a través de página web, independiente de <i>Facebook</i> . Para conectarse a ésta <i>Facebook</i> requerirá el nombre de usuario y contraseña.		

Tabla 48. Requisito software RS-F-002.



Identificador	RS-F-003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-001
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Una vez que esté conectado Usuario a la aplicación, <i>Facebook</i> requerirá unos permisos que el usuario ha de aceptar para que la aplicación pueda acceder a los datos que serán usados durante el desarrollo del juego. Una vez aceptados los permisos no hará falta que vuelvan a ser autorizados cada vez que se ejecute la aplicación. Los permisos solicitados serán sobre:</p> <p><i>Actividades del usuario, cumpleaños del usuario, historial académico del usuario, lugar de nacimiento del usuario, intereses del usuario, lugar de residencia del usuario, relación sentimental del usuario, actividades de los amigos, cumpleaños de los amigos, historial académico de los amigos, lugar de nacimiento de los amigos, intereses de los amigos, lugar de residencia de los amigos, relación sentimental de los amigos.</i></p>		

Tabla 49. Requisito software RS-F-003.



Identificador	RS-F-004		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-001
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Una vez que Usuario se haya conectado a la aplicación y que tenga los permisos aceptados, deberá esperar a que la aplicación realice una llamada FQL (<i>Facebook Query Language</i>) solicitando a las bases de datos de <i>Facebook</i> los datos necesarios para el desarrollo del juego y directamente relacionados con los permisos descritos en la tabla 44. Una vez descargados estos datos se podrá iniciar el juego.</p>		

Tabla 50. Requisito software RS-F-004.

Identificador	RS-F-005		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-001
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Cuando la aplicación haya cargado todos los datos necesarios, ésta activará el botón <i>Empezar juego</i>, el cual tendrá que accionar Usuario para iniciar la partida. Una vez hecho clic sobre dicho botón se cargará toda la interfaz de Usuario.</p>		

Tabla 51. Requisito software RS-F-005.



Identificador	RS-F-006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-002
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Usuario podrá ver, opcionalmente, el tablero de Com en cualquier momento del desarrollo del juego para así poder apreciar los avances que éste realiza. Se producirá un cambio de la interfaz, siendo visible la vista de Com y ocultándose la de Usuario. Usuario siempre podrá volver a su vista produciéndose el efecto contrario.</p>		

Tabla 52. Requisito software RS-F-006.

Identificador	RS-F-007		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-003
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Desde la interfaz de Usuario, éste puede observar la foto de inicio y los datos personales disponibles del amigo-objetivo a descubrir por Com:</p> <p><i>Nombre, sexo, fecha de cumpleaños, lugar de nacimiento y estado sentimental.</i></p> <p>Usuario podrá solicitar ver el resto de los detalles disponibles del objetivo de Com:</p> <p><i>Películas, música, libros, televisión, residencia, historial académico, historial de trabajo, y lugar de nacimiento.</i></p>		

Tabla 53. Requisito software RS-F-007.



Identificador	RS-F-008		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-003
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Desde la interfaz de Usuario, éste puede observar su tablero compuesto por 15 amigos, de los cuales podrá visionar la foto de inicio y los datos personales disponibles:</p> <p><i>Nombre, sexo, fecha de cumpleaños, lugar de nacimiento y estado sentimental.</i></p> <p>Usuario podrá solicitar ver el resto de los detalles disponibles de un amigo en particular de los que forman el tablero:</p> <p><i>Películas, música, libros, televisión, residencia, historial académico, historial de trabajo, y lugar de nacimiento.</i></p>		

Tabla 54. Requisito software RS-F-008.



Identificador	RS-F-009		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-003
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Desde la interfaz de Com, Usuario puede descubrir la foto de inicio y los datos personales disponibles del amigo-objetivo que tiene que adivinar:</p> <p><i>Nombre, sexo, fecha de cumpleaños, lugar de nacimiento y estado sentimental.</i></p> <p>Usuario podrá solicitar ver el resto de los detalles disponibles de su objetivo a descubrir:</p> <p><i>Películas, música, libros, televisión, residencia, historial académico, historial de trabajo, y lugar de nacimiento.</i></p>		

Tabla 55. Requisito software RS-F-009.



Identificador	RS-F-010		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-003
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Desde la interfaz de Com, Usuario puede observar el tablero de Com compuesto por 15 amigos, de los cuales podrá visionar la foto de inicio y los datos personales disponibles:</p> <p><i>Nombre, sexo, fecha de cumpleaños, lugar de nacimiento y estado sentimental.</i></p> <p>Usuario podrá solicitar ver el resto de los detalles disponibles de un amigo en particular de los que forman el tablero de Com:</p> <p><i>Películas, música, libros, televisión, residencia, historial académico, historial de trabajo, y lugar de nacimiento.</i></p>		

Tabla 56. Requisito software RS-F-010.

Identificador	RS-F-011		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-004
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>El usuario podrá consultar un apartado de instrucciones. Estas instrucciones se abrirán en una ventana del navegador independiente a aquella en la que se está ejecutando la aplicación.</p>		

Tabla 57. Requisito software RS-F-011.



Identificador	RS-F-012		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-005
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Usuario podrá formular a Com, por turno, una de las siguientes preguntas acerca de su objetivo:</p> <ol style="list-style-type: none"> 01. ¿Es hombre? 02. ¿Es mujer? 03. ¿Su ciudad de nacimiento es ...? 04. ¿Nació en la comunidad de ...? 05. ¿Nació en el país de ...? 06. ¿Su situación sentimental es soltero(a)? 07. ¿Su situación sentimental es casado(a)? 08. ¿Su situación sentimental es "Es complicado"? 09. ¿Su situación sentimental es "Tiene una relación"? 10. ¿Su situación sentimental es prometido(a)? 11. ¿Su situación sentimental es "Tiene una relación abierta"? 12. ¿Su situación sentimental es viudo(a)? 13. ¿Su situación sentimental es separado(a)? 14. ¿Su situación sentimental es divorciado(a)? 15. ¿Nació en el año ...? 16. ¿Nació en el mes ...? 17. ¿Nació en el día ...? 18. ¿Estudió en el centro ...? 19. ¿Reside en la ciudad de ...? 20. ¿Reside en el país de ...? 21. ¿Trabaja en ...? 22. ¿Uno de sus libros favoritos es ...? 23. ¿Una de sus películas favoritas es ...? 24. ¿Uno de sus artistas/grupos musicales favoritos es ...? 25. ¿Uno de sus programas favoritos de tv es ...? 26. ¿Su nombre es ...? <p>Usuario deberá completar la pregunta en caso de que así fuera necesario.</p>		

Tabla 58. Requisito software RS-F-012.



Identificador	RS-F-013		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-006
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Com podrá realizar una de las siguientes preguntas al azar a Usuario acerca de su objetivo:</p> <ol style="list-style-type: none"> 01. ¿Es hombre? 02. ¿Es mujer? 03. ¿Su ciudad de nacimiento es ...? *03. ¿Tiene especificada su ciudad de nacimiento? 04. ¿Nació en la comunidad de ...? *04. ¿Tiene especificada su comunidad de nacimiento? 05. ¿Nació en el país de ...? *05. ¿Tiene especificado su país de nacimiento? 06. ¿Su situación sentimental es soltero(a)? 07. ¿Su situación sentimental es casado(a)? 08. ¿Su situación sentimental es "Es complicado"? 09. ¿Su situación sentimental es "Tiene una relación"? 10. ¿Su situación sentimental es prometido(a)? 11. ¿Su situación sentimental es "Tiene una relación abierta"? 12. ¿Su situación sentimental es viudo(a)? 13. ¿Su situación sentimental es separado(a)? 14. ¿Su situación sentimental es divorciado(a)? 15. ¿Nació en el año ...? *15. ¿Tiene especificado su año de nacimiento? 16. ¿Nació en el mes ...? *16. ¿Tiene especificado su mes de nacimiento? 17. ¿Nació en el día ...? *17. ¿Tiene especificado su día de nacimiento? 18. ¿Estudió en el centro ...? *18. ¿Tiene especificado dónde estudió o estudia? 19. ¿Reside en la ciudad de ...? *19. ¿Tiene especificada su ciudad de residencia? 20. ¿Reside en el país de ...? *20. ¿Tiene especificado su país de residencia? 21. ¿Trabaja en ...? *21. ¿Tiene especificado dónde trabajó o trabaja? 22. ¿Uno de sus libros favoritos es ...? *22. ¿Tiene especificados sus libros favoritos? 23. ¿Una de sus películas favoritas es ...? *23. ¿Tiene especificadas sus películas favoritas? 24. ¿Uno de sus artistas/grupos musicales favoritos es ...? *24. ¿Tiene especificados grupos o artistas musicales favoritos? 25. ¿Uno de sus programas favoritos de tv es ...? *25. ¿Tiene especificados sus programas de tv favoritos? 26. ¿Su nombre es ...? 		

Tabla 59. Requisito de usuario RS-F-013.



Identificador	RS-F-014		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-006
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Com será capaz de guardar en un Array todas las preguntas que vaya realizando a Usuario, de manera que a la hora de elegir pregunta repase dicho Array comprobando que la pregunta elegida al azar no se encuentre entre las ya realizadas, así no se repetirá pregunta.		

Tabla 60. Requisito software RS-F-014.

Identificador	RS-F-015		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-006
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Si la pregunta elegida al azar que va a realizar Com a Usuario necesita ser completada, Com seleccionará a un amigo al azar de los que se encuentran en su tablero, y que no haya sido descartado previamente, para utilizar el campo correspondiente con el fin de completar la pregunta.		

Tabla 61. Requisito software RS-F-015.



Identificador	RS-F-016		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-007
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Usuario podrá responder a las preguntas formuladas por Com con las siguientes opciones: <ul style="list-style-type: none"> • Si • No • Sin detalles 		

Tabla 62. Requisito software RS-F-016.

Identificador	RS-F-017		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-008
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Com responderá a las preguntas formuladas por Usuario en función del amigo-objetivo de este último. Las posibles respuestas: <ul style="list-style-type: none"> • Si • No • Sin detalles 		

Tabla 63. Requisito software RS-F-017.



Identificador	RS-F-018		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-009
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Usuario podrá descartar de su tablero a todos aquellos amigos/as que no cumplan la respuesta a la pregunta que realizó con anterioridad a Com. La interfaz mostrará que un amigo del tablero de Usuario ha sido descartado.		

Tabla 64. Requisito software RS-F-018.

Identificador	RS-F-019		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-010
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Com podrá descartar de su tablero, con total exactitud, a todos aquellos amigos que no cumplan la respuesta a la pregunta que realizó con anterioridad a Usuario. La interfaz mostrará que un amigo del tablero de Com ha sido descartado.		

Tabla 65. Requisito software RS-F-019.



Identificador	RS-F-020		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-011
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Usuario podrá resolver al principio de su turno, y cuando considere oportuno durante el desarrollo de la partida, seleccionando de su tablero al candidato que crea ser su objetivo a identificar.		

Tabla 66. Requisito software RS-F-020.

Identificador	RS-F-021		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-012
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Com, al principio de su turno, comprobará cuantos amigos quedan sin descartar de su tablero. En caso de que sólo quede uno resolverá ganando la partida.		

Tabla 67. Requisito software RS-F-021.



Identificador	RS-F-022		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-013
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Usuario podrá salir del juego en cualquier momento desconectándose de <i>Facebook</i> .		

Tabla 68. Requisito software RS-F-022.

Identificador	RS-F-023		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-C-014
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Usuario podrá ceder el turno a Com cuando considere que haya terminado de descartar.		

Tabla 69. Requisito software RS-F-023.



5.3.1.2 *Requisitos de Interfaz*

Identificador	RU-I-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-010
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Asegurar la correcta visualización de la aplicación en los navegadores <i>Microsoft Internet Explorer®</i> , <i>Mozilla Firefox®</i> y <i>Google Chrome®</i> .		

Tabla 70. Requisito de usuario RU-I-001.



5.3.1.3 Requisitos de operación

Identificador	RS-O-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-002
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Cada participante del juego, Usuario y Com, ha de tener un tablero, oculto para su adversario, compuesto por 15 amigos/as seleccionados al azar de entre los que tiene el usuario en su cuenta de <i>Facebook</i> . El tablero mostrará las fotos de inicio y datos personales de cada amigo.		

Tabla 71. Requisito software RS-O-001.

Identificador	RS-O-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-003
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Cada participante del juego, Usuario y Máquina, ha de tener un objetivo, oculto para su adversario, compuesto por un amigo/a seleccionado al azar de entre todos los que se encuentran en el tablero de su oponente. La interfaz mostrará la foto y datos personales del objetivo en la vista correspondiente.		

Tabla 72. Requisito software RS-O-002.



Identificador	RS-O-003		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-004
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Disponer de una interfaz gráfica lo más sencilla e intuitiva posible, que propicie una partida fluida.		

Tabla 73. Requisito software RS-O-003.

Identificador	RS-O-004		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-004
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>La interfaz contará con un apartado de controles en la vista de Usuario que se activarán en el momento que éste decida comenzar la partida. Los controles, y su presentación, permitirán un manejo intuitivo de la aplicación. Habrá tres presentaciones de controles:</p> <ul style="list-style-type: none"> • La primera presentación ofrecerá unos controles que permitirán a Usuario realizar preguntas a Com, ver la respuesta de éste, descartar a los amigos del tablero o ir a la segunda presentación, donde Usuario puede resolver. • La segunda es la que da la opción a Usuario de resolver. Podrá resolver, o por el contrario, volver a la primera presentación. • La última es la presentación donde Com realizará la pregunta y Usuario le dará la respuesta. 		

Tabla 74. Requisito software RS-O-004.



Identificador	RS-O-005		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-004
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>En el apartado de controles de la interfaz, las presentaciones se irán sucediendo en función de las acciones que realice Usuario, variando dicha interfaz en cada presentación, de manera que se obtiene un manejo fluido de la aplicación. Las sucesiones se producirán del siguiente modo:</p> <ul style="list-style-type: none"> De la primera presentación se irá a la segunda si Usuario decide resolver. En la primera presentación Usuario puede hacer una pregunta. Después recibirá la respuesta y hará el descarte. Cuando haya finalizado el descarte pasará a la tercera presentación. De la segunda presentación el Usuario puede volver a la primera. En la tercera presentación Com realizará la pregunta. Inmediatamente después de que Usuario de la respuesta la aplicación ocultará la tercera presentación, mostrando la primera. 		

Tabla 75. Requisito software RS-O-005.



Identificador	RS-O-006		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-005
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Cuando Usuario y Com decidan descartar a un amigo de sus respectivos tableros la aplicación ocultará todos los datos de ese amigo y sustituirá su nombre por la palabra <i>DESCARTADO</i>.</p>		

Tabla 76. Requisito software RS-O-006.

Identificador	RS-O-007		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-006
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>La aplicación debe recoger y mostrar todos los detalles personales y foto de inicio aparecidos en <i>Facebook</i> de un amigo/a del usuario que participe en el juego, tanto si forma parte del tablero como si pertenece a uno de los objetivos. Estos detalles personales serán:</p> <p><i>Nombre, sexo, fecha de cumpleaños, lugar de nacimiento y estado sentimental.</i></p> <p>Usuario podrá solicitar a la aplicación observar más detalles disponibles correspondientes a un determinado amigo que desea consultar, mostrándolos ésta por pantalla. Estos datos pueden ser:</p> <p><i>Películas, música, libros, televisión, residencia, historial académico, historial de trabajo, y lugar de nacimiento.</i></p>		

Tabla 77. Requisito software RS-O-007.



Identificador	RS-O-008		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-007
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>Cuando la aplicación haya encontrado un ganador, mostrará un mensaje en el cual informará a Usuario quién ha sido el vencedor de la partida.</p>		

Tabla 78. Requisito software RS-O-008.

Identificador	RS-O-009		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-008
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	<p>A cada presentación de los controles le corresponderá unos mensajes de ayuda dando consejos acerca de lo que tiene que hacer el usuario en todo momento.</p> <p>Hay dos tipos de mensajes, el <i>título</i> y el <i>pie</i> de presentación. El <i>título</i> dará una idea general de la acción a realizar por el Usuario y el <i>pie</i> de presentación dará alguna indicación o detallará la acción a realizar.</p>		

Tabla 79. Requisito software RS-O-009.



Identificador	RS-O-010		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	RU-R-009
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	La aplicación será capaz de volver a cargar de <i>Facebook</i> todos los amigo/as, tanto del objetivo como del tablero, de Usuario si éste no acertó a la hora de resolver su objetivo.		

Tabla 80. Requisito software RS-O-010.



5.3.1.4 Requisitos de documentación

Identificador	RS-D-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo de Desarrollo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Se proporcionará un manual de usuario impreso al cliente en el que se incluya una descripción detallada del manejo de la aplicación. El idioma del manual será el castellano.		

Tabla 81. Requisito software RS-D-001.



5.3.1.5 Requisitos de calidad

Identificador	RS-C-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo de Desarrollo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Comprobar la adecuación de todos los documentos entregados a lo largo del proyecto al estándar establecido para este desarrollo software.		

Tabla 82. Requisito software RS-C-001.

Identificador	RS-C-002		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo de Desarrollo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Realizar copias de seguridad cuando se alcancen ciertos hitos o logros.		

Tabla 83. Requisito software RS-C-002.



5.4 Análisis de los casos de uso

En este apartado se realiza un estudio de los casos de uso identificados en el estudio de Viabilidad del Sistema con el fin de identificar las clases necesarias para el sistema así como las relaciones entre ellas y su asociación.

El objetivo es identificar el mínimo número de clases que serán necesarias para el sistema, pero suficientes para cubrir todos los requisitos especificados.

5.4.1 Identificación de las clases asociadas a un caso de uso

En la siguiente tabla se recogen las relaciones de interacción entre los casos de uso recogidos durante la fase de Estudio de Viabilidad y las clases de las que se compone el sistema.

Identificador	Clases asociadas
CU-001	<ul style="list-style-type: none">• Aplicación• Interfaz Usuario
CU-002	<ul style="list-style-type: none">• Interfaz Usuario• Interfaz Com
CU-003	<ul style="list-style-type: none">• Interfaz Usuario• Instrucciones
CU-004 CU-005 CU-006 CU-007 CU-008	<ul style="list-style-type: none">• Interfaz Usuario
CU-009	<ul style="list-style-type: none">• Interfaz Usuario• Interfaz Com• Aplicación
CU-010 CU-011	<ul style="list-style-type: none">• Interfaz Usuario
CU-012 CU-013	<ul style="list-style-type: none">• Interfaz Com
CU-014 CU-015	<ul style="list-style-type: none">• Interfaz Usuario

CU-016	<ul style="list-style-type: none"> Interfaz Com
CU-017	<ul style="list-style-type: none"> Interfaz Usuario

Tabla 84. Clases asociadas a casos de uso.

5.4.2 Descripción de la interacción de objetos

En este apartado se muestran los diagramas de secuencia que representan las interacciones entre las clases descritas en el apartado anterior en cada uno de los casos de uso. Se muestran los diagramas más significativos sobre el funcionamiento del sistema

- CU-001 Iniciar Aplicación: Usuario quiere iniciar una partida, para ello deberá iniciar *Aplicación*, figura 58. Una vez que esté iniciada, Usuario podrá dar comienzo al juego, entonces *Aplicación* iniciará la *InterfazUsuario*, la cual permite el desarrollo del juego.

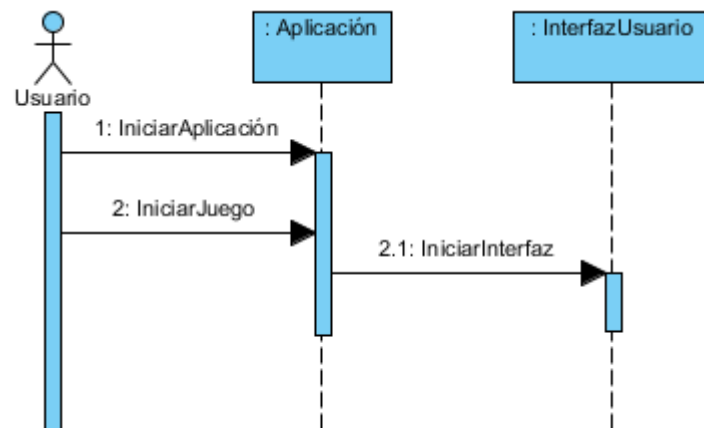


Figura 58. Diagrama de secuencia. Caso de uso CU-001. Iniciar juego.

- CU-002 Cambiar Vista: Usuario seleccionará en la *InterfazUsuario* que desea cambiar de vista, queriendo observar el tablero de Com. *InterfazUsuario* se ocultará e iniciará la vista Com, figura 59. Una vez que Usuario haya terminado de consultar la vista de Com, podrá solicitar a *InterfazCom* volver a la *InterfazUsuario*, haciéndose de nuevo visible y ocultándose la vista de Com.

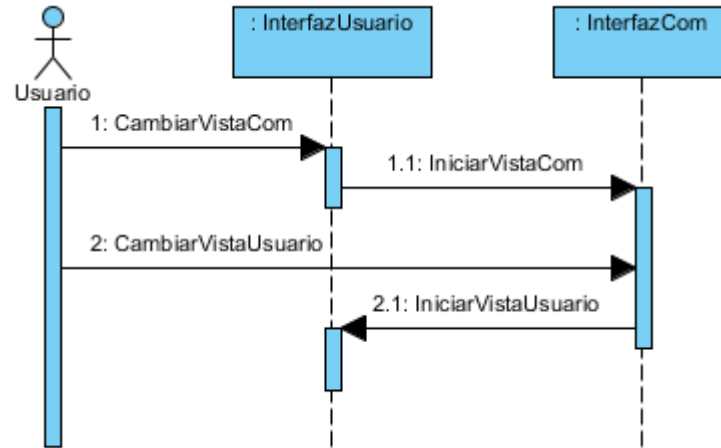


Figura 59. Diagrama de secuencia. Caso de uso CU-002. Cambiar vista.

- CU-003 Ver instrucciones: Usuario podrá solicitar a *InterfazUsuario* consultar las instrucciones y reglas del juego, figura 60. *InterfazUsuario* lanzará *Instrucciones*, abriéndose esta en una nueva ventana del navegador.

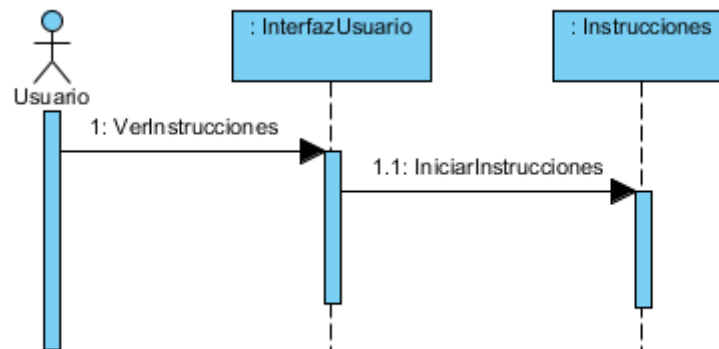


Figura 60. Diagrama de secuencia. Caso de uso CU-003. Ver instrucciones.

- CU-004 Realizar pregunta a Com: Usuario confeccionará una pregunta de entre las que hay disponibles en la *InterfazUsuario* y se la enviará a Com, figura 61.

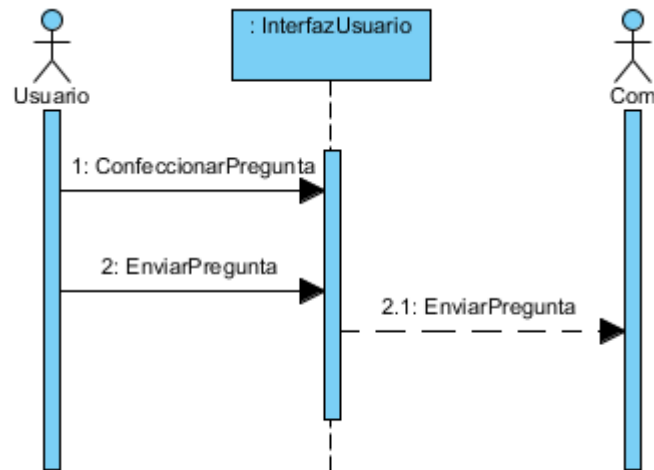


Figura 61. Diagrama de secuencia. Caso de uso CU-004. Realizar pregunta a Com.

- CU-005 Contestar pregunta de Com: Usuario podrá leer la pregunta realizada por Com en la *InterfazUsuario*, figura 62, este dará su respuesta a Com a través de esta clase.

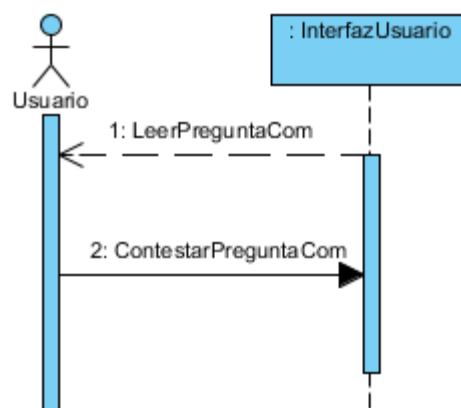


Figura 62. Diagrama de secuencia. Caso de uso CU-005. Contestar pregunta de Com

- CU-006 Descartar amigos/as tablero usuario: Usuario leerá la respuesta de Com a su pregunta, en función de esta seleccionará a los amigos/as de su tablero y posteriormente los descartará, figura 63. *InterfazUsuario* modificará el tablero mostrando cuales han sido descartados.

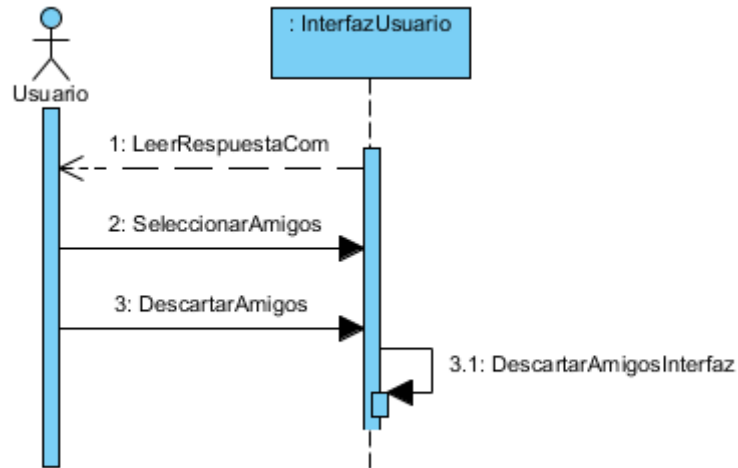


Figura 63. Diagrama de secuencia. Caso de uso CU-006. Descartar amigos/as tablero usuario.

- CU-008 Resolver Usuario: Usuario elige resolver antes que formular una pregunta. Selecciona del tablero *InterfazUsuario* al amigo que considere ser la solución, figura 64. Entonces elegirá resolver, *InterfazUsuario* comprobará si la elección ha sido la correcta y mostrará a Usuario que ha ganado.

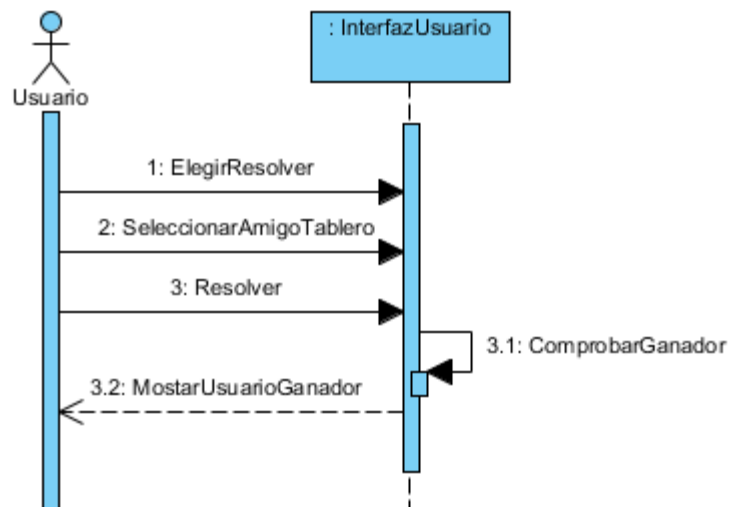


Figura 64. Diagrama de secuencia. Caso de uso CU-008. Resolver Usuario.

- CU-008 Resolver Usuario. Escenario Alternativo: El procedimiento es igual que en el caso anterior, pero Usuario no ha acertado en su elección e *InterfazUsuario* volverá a cargar el tablero de manera que usuario empiece de nuevo la partida, figura 65. Finalmente, Usuario cederá el turno a Com.

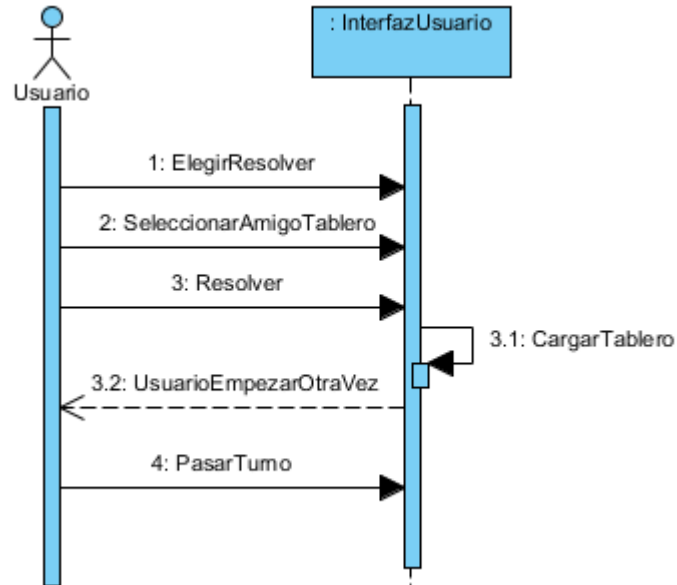


Figura 65. Diagrama de secuencia. Caso de uso CU-008. Resolver Usuario. Escenario Alternativo.

- CU-014 Realizar pregunta a Usuario: Usuario cederá su turno a Com, figura 66, este automáticamente realizará una pregunta a Usuario.

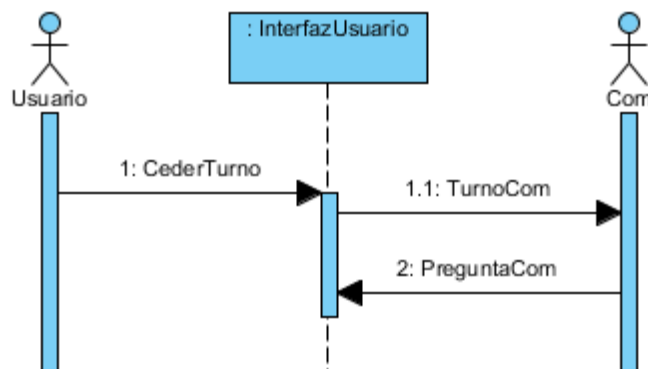


Figura 66. Diagrama de secuencia. Caso de uso CU-014. Realizar pregunta a Usuario.

- CU-016 Realizar pregunta a Usuario: *InterfazUsuario* muestra a Com la respuesta que Usuario a dado a su pregunta, Com, en función de esta, descarta de su tablero en la *InterfazCom* aquellos amigos/as que no cumplen la respuesta. Finalmente Com cede el turno, figura 67.

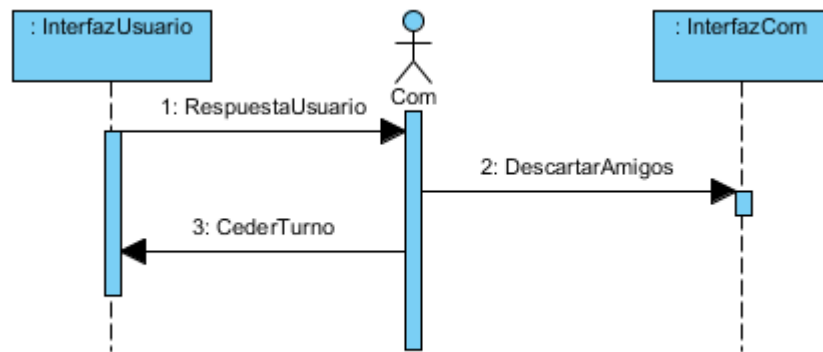


Figura 67. Diagrama de secuencia. Caso de uso CU-016. Realizar pregunta a Usuario.

5.5 Análisis de clases

En este apartado se realizará una descripción detallada de cada una de las clases identificadas en el proceso de análisis. Además se definirán sus atributos y las relaciones existentes entre ellas.

Puesto que todavía se está en la fase de análisis, el modelo obtenido aquí será un análisis conceptual de lo que el cliente requiere, y será en la siguiente etapa del proyecto, en la fase de diseño, en la que se realizará un estudio más exhaustivo.

En la figura 68 se muestra el diagrama de clases resultante y que se explicará en los posteriores subapartados.

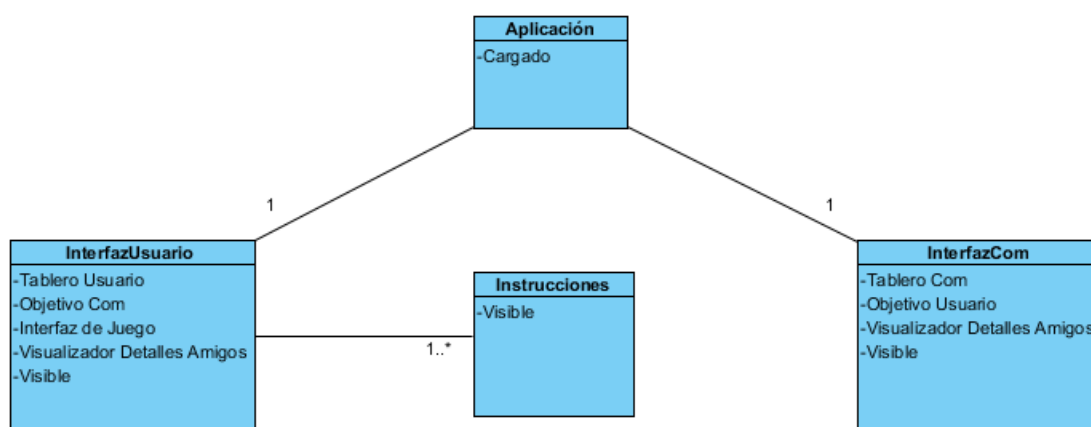


Figura 68. Diagrama conceptual de clases.



5.5.1 Identificación de responsabilidades y atributos

En este apartado se especifican las responsabilidades asociadas a cada una de las clases representadas en el diagrama de la figura 58 describiendo el papel que desempeñan los objetos pertenecientes a cada una de ellas en los diferentes casos de uso del sistema. Además, se describen los atributos correspondientes a las clases, los cuales especifican propiedades de éstas y se involucran en las diferentes responsabilidades de la clase. Toda esta información queda recogida en las tablas 85-88.

Clase	Aplicación
Responsabilidades	La función de esta clase es la de cargar las interfaces, encargarse de la IA y realizar las consultas a las BB.DD. de <i>Facebook</i> para obtener los datos que después serán utilizados en el juego.
Atributos	<ul style="list-style-type: none">• Cargado: Aplicación esperará a tener cargadas las interfaces, la IA y los datos necesarios para poder empezar el juego.

Tabla 85. Clase Aplicación.

Clase	InterfazUsuario
Responsabilidades	Clase que contiene el tablero Usuario y los controles que permiten a éste manejar el juego.
Atributos	<ul style="list-style-type: none">• TableroUsuario: contiene a 15 amigos/as del Usuario conectado a la aplicación así como los detalles personales de éstos.• ObjetivoCom: amigo/a del Usuario conectado a la aplicación y que ha de ser descubierto por Com.• InterfazdeJuego: contendrá los botones y controles del juego que permiten al Usuario poder desarrollar la partida.• VisualizadorDetallesAmigos: en esta área de la interfaz Usuario podrá ver más detalles tanto de los amigos que conforman el tablero como del objetivo a descubrir por Com.• Visible: este atributo indica si esta interfaz es visible para Usuario o por el contrario está oculta en detrimento de la otra interfaz.

Tabla 86. Clase InterfazUsuario.



Clase	InterfazCom
Responsabilidades	Clase que contiene el tablero de Com y los controles que permiten observar los detalles de los amigos que forman el tablero.
Atributos	<ul style="list-style-type: none">• TableroCom: contiene a 15 amigos/as del Usuario conectado a la aplicación y los detalles personales de éstos.• ObjetivoUsuario: amigo/a del Usuario conectado a la aplicación, el cuál ha de ser descubierto por éste.• VisualizadorDetallesAmigos: en esta área de la interfaz Usuario podrá ver más detalles tanto de los amigos que conforman el tablero de Com como del objetivo a descubrir por sí mismo.• Visible: este atributo indica si esta interfaz es visible para Usuario o por el contrario está oculta en detrimento de la otra interfaz.

Tabla 87. Clase InterfazCom.

Tabla xx. Clase InterfazCom.

Clase	Instrucciones
Responsabilidades	Clase que permite al Usuario poder observar las instrucciones y reglas para un buen desarrollo de la partida.
Atributos	<ul style="list-style-type: none">• Visible: este atributo indica si las instrucciones son visibles para Usuario.

Tabla 88. Clase Instrucciones.



5.5.2 Identificación de asociaciones

En esta tarea se estudiarán las relaciones de asociación entre las distintas clases identificadas en el diagrama de la figura 68.

Cada una de estas asociaciones será especificada con su cardinalidad, con el número de instancias que pueden estar relacionadas con una instancia del otro lado de la asociación, y con una breve descripción de su funcionalidad. Esta información queda reflejada en las tablas 89-91.

No se realizará el estudio de generalizaciones debido a que no hay clases que compartan estructura y/o comportamiento con otras clases.

Clases asociadas	Aplicación-InterfazUsuario
Cardinalidades	<ul style="list-style-type: none">• Aplicación (1)• InterfazUsuario (1)
Descripción	La aplicación carga los datos necesarios para el desarrollo de la partida y, una vez hecho esto, permite lanzar la interfaz de Usuario para dar comienzo así a la partida.

Tabla 89. Asociación clases Aplicación e InterfazUsuario

Clases asociadas	Aplicación-InterfazCom
Cardinalidades	<ul style="list-style-type: none">• Aplicación (1)• InterfazCom (1)
Descripción	La aplicación carga los datos necesarios para el desarrollo de la partida y, una vez hecho esto, permite lanzar la interfaz de Com, pudiendo ser consultada por Usuario durante el desarrollo de la partida siempre que él lo desee.

Tabla 90. Asociación clases Aplicación e InterfazCom

Clases asociadas	InterfazUsuario-Instrucciones
Cardinalidades	<ul style="list-style-type: none">• InterfazUsuario (1)• Instrucciones (1..*)
Descripción	InterfazUsuario permitirá que en cualquier momento se consulten las instrucciones del juego.

Tabla 91. Asociación clases InterfazUsuario e Instrucciones.



5.6 Definición de las interfaces de usuario

Uno de los aspectos más importantes de una aplicación con la que interactúan los usuarios es la interfaz, ya que es el medio de comunicación entre el usuario y el sistema.

5.6.1 Especificación de principios generales de la interfaz

En este proyecto se ha de contar con una interfaz intuitiva y que permita al usuario poder desarrollar partidas de manera fluida y sin contratiempos.

Debido a que la aplicación se trata de un juego, el perfil del usuario puede variar mucho entre uno y otro, por lo que es recomendable que la interfaz sea lo más clara e intuitiva posible. De manera que el usuario después de una partida este adaptado por completo a la interfaz de ésta.

Con el objetivo de facilitar el proceso de adaptación de los usuarios a la aplicación, además de ser lo más sencillas posibles, se tratará de proporcionar información precisa acerca de las acciones que debe ir realizando el usuario de la aplicación.

5.6.2 Identificación de perfiles y diálogos

Existirá un único perfil de usuario que deberá estar registrado en la red social *Facebook*.

5.6.3 Especificación de formatos individuales de la interfaz de la pantalla

A continuación se muestran las interfaces de usuario para la aplicación desarrollada, describiendo los elementos que la componen.

5.6.3.1 Interfaz de la aplicación

Es la interfaz inicial, es decir, la que le aparecerá al usuario una vez que se haya conectado tanto a *Facebook* como a la aplicación.



Figura 69. Interfaz de la aplicación.

El aspecto de la misma se puede ver en la figura 69. En ella se pueden apreciar los siguientes elementos:

- **Botón *Empezar juego*:** la aplicación no dará funcionalidad a este botón hasta que hayan sido cargados todos los datos que necesite ésta para el desarrollo del juego.
- **Botón *Ir a vista Com*:** este botón de momento está desactivado. Una vez que haya sido activado, permitirá al usuario de la aplicación ocultar la interfaz Usuario y hacer la visible la interfaz Com.
- **Botón *Salir*:** este botón permite al usuario salir de la aplicación y desconectarse de *Facebook*.
- **Tablero Usuario:** Será el lugar donde la aplicación empezará a cargar los datos necesarios para el funcionamiento del juego como es el caso del tablero de usuario (cuadro remarcado en azul de la figura 69), el cual está compuesto por 15 amigos/as que tenga agregados a su cuenta de *Facebook* el usuario de la aplicación. Cada amigo vendrá compuesto por su foto de perfil, además de una



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

serie de datos que el amigo podrá tener, o no, especificados. Estos datos pueden ser: nombre, sexo, lugar de nacimiento, fecha de nacimiento y situación sentimental.

5.6.3.2 Interfaz Usuario

Una vez que la aplicación haya obtenido todos los datos que necesitaba y que el usuario haya pulsado el botón *Empezar juego*, se cargará la interfaz Usuario, figura 70.



Figura 70. Interfaz Usuario.

En esta interfaz se pueden diferenciar varias partes:

1. **Controles superiores:** Parte superior de la interfaz, figura 71, en la que se puede observar el nombre de la aplicación y tres botones que no influyen directamente en el desarrollo de la partida:

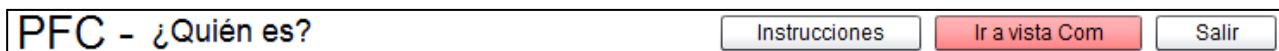


Figura 71. Interfaz Usuario. Parte superior.

- **Botón Instrucciones:** Este botón permite al usuario poder consultar las instrucciones y reglas del juego.
- **Botón Ir a vista Com:** a diferencia de la interfaz de la aplicación, el botón ya está activado. Este botón permite al usuario ocultar la interfaz en la que se encuentra, interfaz Usuario, y hace visible la interfaz Com.

- **Botón *Salir*:** Al igual que en la interfaz de la aplicación, este botón permite salir de la aplicación y desconectarse de *Facebook*.
2. **Tablero Usuario:** descrito anteriormente en el punto 5.7.3.1 *Interfaz Aplicación*.
 3. **Objetivo Com y visualizador de detalles:** Es el recuadro que se encuentra en la esquina inferior izquierda, figura 72. A continuación se detalla esta zona de la interfaz:

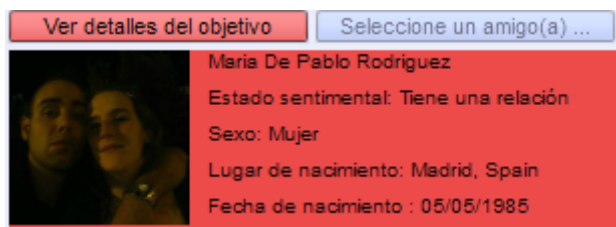


Figura 72. Interfaz Usuario. Objetivo Com y visualizador de detalles.

- **Objetivo Com:** por defecto se podrá observar el amigo a descubrir por el oponente (Com). Vendrá compuesto, al igual que los amigos en el tablero, por su foto de perfil y los siguientes datos que tenga disponible: nombre, sexo, lugar de nacimiento, fecha de nacimiento y situación sentimental.
- **Botón *Ver detalles del Objetivo*:** Este botón permitirá al Usuario poder consultar el resto de datos que pueda tener especificados el amigo objetivo en la red social. Los datos que podrían observarse son: centros de enseñanza donde cursó estudios, residencia actual, empresas o centros en los que ha trabajado y libros, programas de televisión, películas y artistas musicales favoritos. En la figura 73 se muestra un ejemplo de visualización de estos datos extra.

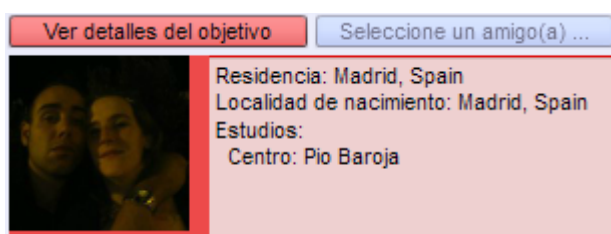


Figura 73. Interfaz Usuario. Objetivo Com y visualizador de detalles. Visualizando detalles Objetivo.

- **Botón *Seleccione un amigo(a)...***: Este botón permite al Usuario poder observar los detalles de alguno de los amigos de los que componen su tablero de juego. Si pincha sobre un amigo del tablero, el título del botón cambiará, siendo ahora su título “Ver detalles de + *amigo seleccionado*”. Se puede observar el nombre completo poniendo el cursor sobre el botón. Un ejemplo de esto aparece en la figura 74.

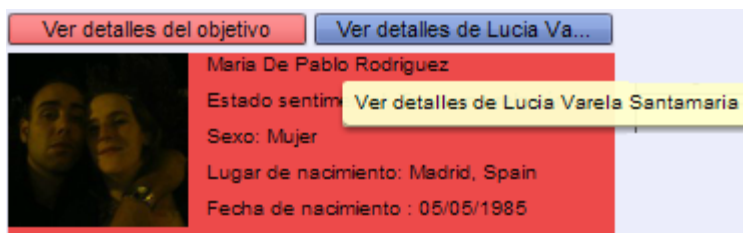


Figura 74. Interfaz Usuario. Objetivo Com y visualizador de detalles. Seleccionando amigo del tablero.

- Cuando se pulsa el botón anterior aparecerán los detalles del amigo seleccionado, figura 75, y para volver a observar al objetivo simplemente habrá que seleccionar otro amigo del tablero para que aparezca automáticamente.

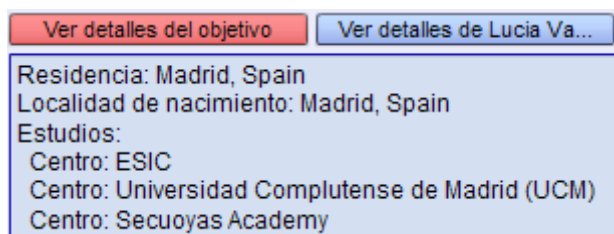


Figura 75. Interfaz Usuario. Objetivo Com y visualizador de detalles. Observando detalles del amigo.

4. **Controles del juego:** estos controles pueden observarse en la esquina inferior derecha de la interfaz, figura 76. Dependiendo de a quién pertenezca el turno, se podrán distinguir varias etapas en las que los controles variarán para poder avanzar en el desarrollo del juego logrando una partida fluida.

Hay una característica que todas las etapas tienen en común: el título que recibe la etapa, la cual ya indica la acción a realizar por Usuario, y los mensajes de ayuda que aparecen en la parte inferior de los controles del juego, que ayudan a que la partida sea más intuitiva.

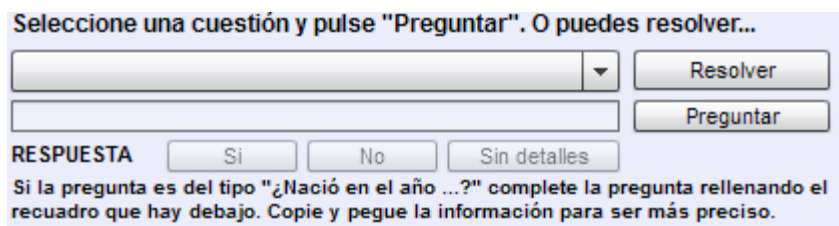


Figura 76. Interfaz Usuario. Controles del juego.

Dependiendo de turno, las etapas son:

- **Turno Usuario:** En su turno Usuario podrá realizar una pregunta, ver la respuesta de Com, descartar a los amigos que no cumplan dicha respuesta, o resolver. Usuario tendrá varias etapas para realizar estas acciones:

1. **Etapla pregunta/resolver:** Se trata de la primera etapa. Aquí el usuario deberá decidir entre resolver o confeccionar una pregunta. Esta etapa está compuesta de los siguientes aspectos, figura 77:

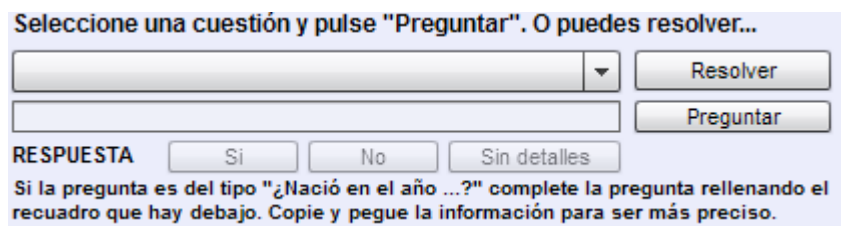


Figura 77. Interfaz Usuario. Controles del juego. Turno Usuario. Etapla preguntar/resolver.

- **Barra de selección de pregunta:** aquí se encontrarán todas las preguntas que se pueden realizar a Com.
 - **Cuadro de entrada de texto:** hay preguntas que deben ser completadas. Para ello se utilizará este cuadro para introducir el texto que la complete.
 - **Respuesta:** en esta etapa están desactivados los botones de respuesta.
 - **Botón Resolver:** permite al usuario ir a la etapa de resolver.
 - **Botón Preguntar:** una vez que Usuario tenga su pregunta lista, dando a este botón le enviará la pregunta a Com para que la responda.
2. **Etapla respuesta/descartar:** esta etapa dará comienzo en el instante que usuario presione el botón *Preguntar*. Entonces se hará visible el botón *Descartar* y la respuesta de Com a la pregunta realizada, figura 78. Los componentes serán los siguientes:

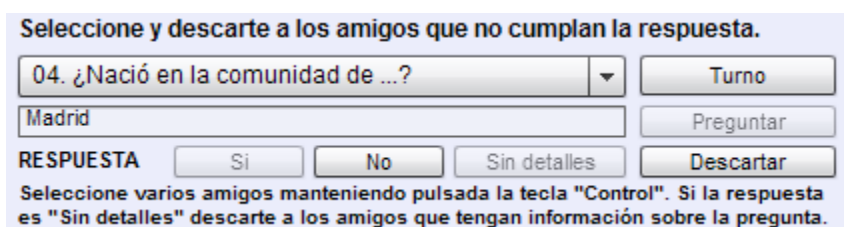


Figura 78. Interfaz Usuario. Controles del juego. Turno Usuario. Etapla respuesta/descartar.

- **Respuesta:** se puede observar la respuesta de Com a la pregunta realiza.
- **Botón Descartar:** usuario deberá descartar a todos los amigos del tablero que no cumplan la respuesta. Para ello seleccionará los amigos/as correspondientes y presionará sobre el botón descartar.

- **Botón *Turno*:** este botón dará fin a la etapa de descartar cediendo el turno a Com.

3. **Etapa ceder turno:** una vez que Usuario haya descartado se podrá ceder el turno a Com para que realice su pregunta. Como se ve en la figura 79, en esta etapa sólo cambia el título de la etapa y su mensaje de ayuda correspondiente.

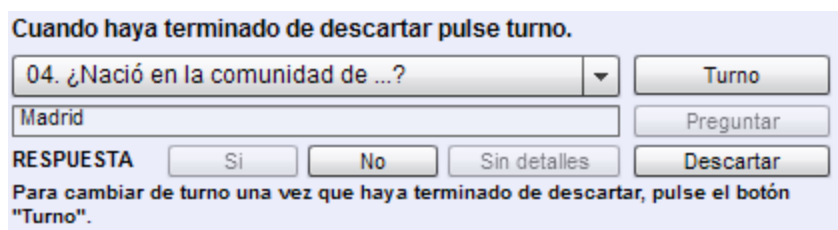


Figura 79. Interfaz Usuario. Controles del juego. Turno Usuario. Etapa ceder turno.

4. **Etapa resolver:** al principio del turno, usuario elige si resolver o realizar una pregunta. En caso de querer resolver pulsará el botón de dicho nombre apareciendo la figura 80. En ella se encontrarán los dos siguientes botones:

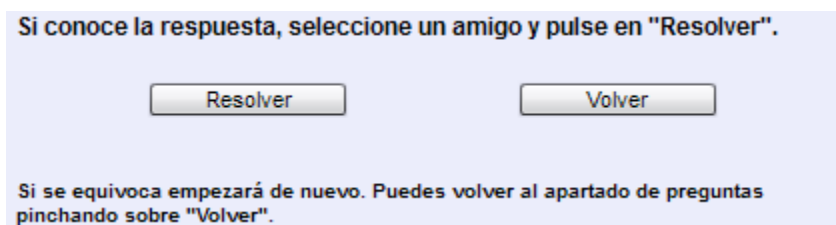


Figura 80. Interfaz Usuario. Controles del juego. Turno Usuario. Etapa resolver.

- **Botón *Resolver*:** Usuario seleccionará un amigo de entre todos los que forman el tablero y pulsará resolver. En ese instante sabrá si ha ganado o si, por el contrario, ha de empezar de nuevo.
- **Botón *Volver*:** permite al Usuario volver a la etapa anterior.

5. **Etapa empezar de nuevo:** a la hora de resolver, los controles de juego mostrarán la figura 81 si Usuario se ha equivocado. En ella se observe un único botón:

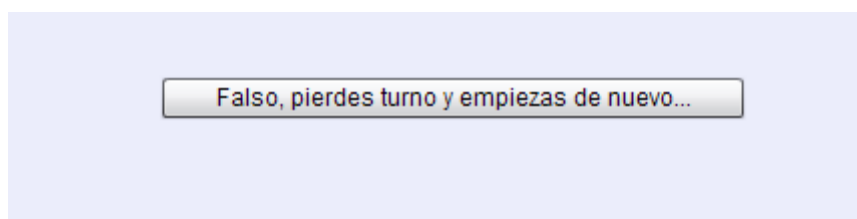


Figura 81. Interfaz Usuario. Controles del juego. Turno Usuario. Etapa empezar de nuevo.

- **Botón *Falso, pierdes turno y empiezas de nuevo...***: como bien indica su nombre, da a conocer a Usuario que no ha acertado, y que tiene que empezar de cero, además de ceder el turno a Com.
- **Turno Com**: En su turno, Com podrá realizar una pregunta, recibir la respuesta de Usuario, descartar a los amigos que no cumplan dicha respuesta o resolver. El turno de Com solo consta de una etapa:

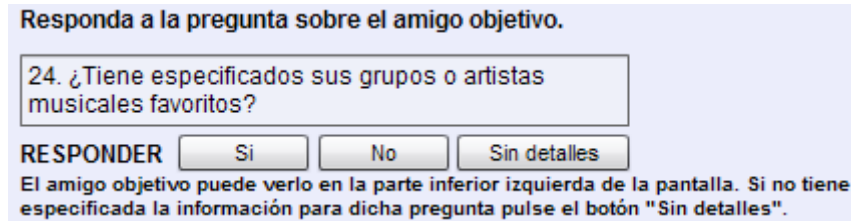


Figura 82. Interfaz Usuario. Controles del juego. Turno Com. Etapa pregunta/respuesta

1. **Etapa pregunta/respuesta**: Se trata de la única etapa, figura 82. Aquí Com lanzará su pregunta quedándose a la espera de que Usuario dé su contestación.
 - **Cuadro de salida de texto**: se puede observar la pregunta de Com.
 - **Botones *Responder***: usuario consultará los datos del amigo/objetivo de Com y pulsará el botón correspondiente.

Como se puede observar, la duración de este turno es más corta, debido a que tanto la pregunta como el descarte lo realiza la IA de Com automáticamente.

5.6.3.3 Interfaz Com

Si el usuario que está utilizando la aplicación desea consultar los avances de Com, puede en cualquier momento consultar la vista de Com, ocultando la suya propia, de modo que puede observar el tablero de Com y consultar los detalles de los amigos que lo forman. También, si así lo desea Usuario, éste puede descubrir su objetivo en esta vista.



Figura 83. Interfaz Com.

Esta interfaz, figura 83, también está dividida en partes:

1. **Controles superiores:** en la zona superior se pueden encontrar dos botones, figura 84:



Figura 84. Interfaz Com. Controles superiores.

- **Botón Ir a vista Player:** este botón, al hacer clic sobre él, permite volver a la interfaz Usuario, pudiendo continuar con el juego.
- **Botón Salir:** al igual que el botón de mismo nombre de la interfaz Usuario, permite salir de la aplicación y de Facebook.

2. **Tablero Com:** aquí Usuario puede consultar los 15 amigos que han sido seleccionados para formar el tablero de Com y los avances que realiza éste durante el juego, figura 85. También puede observar que entre todos los amigos/as se encuentra el amigo-objetivo de Com que puede ver en su interfaz Usuario.

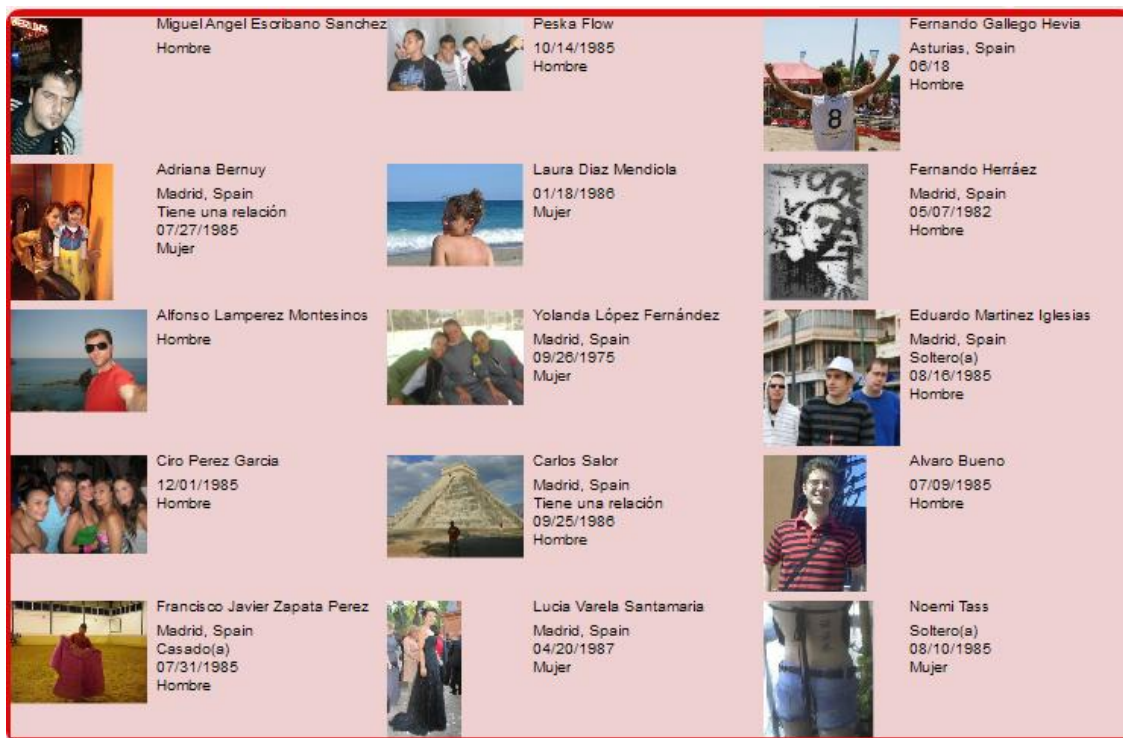


Figura 85. Interfaz Com. Tablero Com.

3. **Objetivo Usuario y visualizador de detalles:** se encuentra en la parte inferior de la interfaz y puede dividirse en dos secciones, figura 86. Una donde se podrá ver el objetivo propio a descubrir y otra donde visualizar los detalles, incluidos los de los amigos/as que componen el tablero de Com.

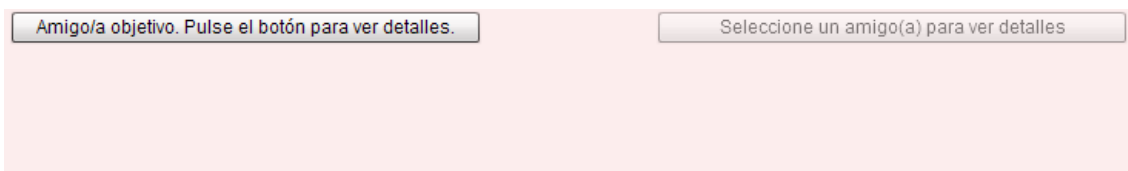


Figura 86. Interfaz Com. Objetivo Usuario y visualizador de detalles.

- **Botón Amigo/a objetivo. Pulse el botón para ver detalles:** si Usuario decide descubrir su objetivo antes de averiguarlo por métodos propios, debe utilizar este botón, de manera que quedará a la vista su objetivo, con su foto de perfil y sus datos personales, además de poder observarse el resto de detalles en un cuadro de texto que se puede encontrar en la parte inferior derecha, figura 87.

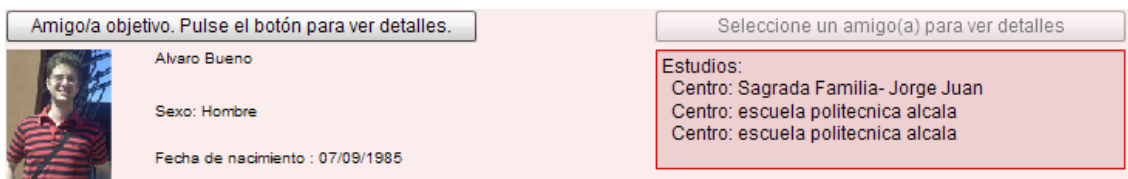


Figura 87. Interfaz Com. Objetivo Usuario y visualizador de detalles. Ver Objetivo Usuario.

- **Botón *Seleccione un amigo/a para ver detalles:*** Usuario, utilizando este botón, podrá observar los detalles de cualquier amigo seleccionándolo previamente del tablero de Com. La información aparecerá justo debajo del botón dentro de un cuadro de texto, figura 88.



Figura 88. Interfaz Com. Objetivo Usuario y visualizador de detalles. Ver detalles amigo/a

5.7 Análisis de consistencia y especificación de requisitos

En esta sección se detalla cómo se asegurará la calidad entre los distintos modelos que se han generado como productos, comprobando la consistencia entre ellos.

5.7.1 Verificación y análisis de consistencia entre modelos.

Mediante esta tarea se comprobará que cada modelo generado es correcto comprobando la coherencia de los modelos y la existencia de ambigüedad o redundancia.

Las diferentes comprobaciones varían dependiendo del tipo de desarrollo, aunque generalmente son matrices entre los elementos comunes de los distintos modelos. Para este desarrollo se ha elegido matrices de trazabilidad que permiten asegurar la consistencia existente entre el Estudio de Viabilidad del Sistema y el Análisis.

5.7.1.1 Matriz de Requisitos de usuario de capacidad-Casos de uso.

En esta matriz se observa la correspondencia entre los casos de uso y los requisitos de capacidad recogidos en la fase de Estudio de Viabilidad del Sistema, observando la coherencia entre unos casos y otros.

	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013	CU-014	CU-015	CU-016	CU-017
RU-C-001	X																
RU-C-002		X															
RU-C-003										X	X	X	X				
RU-C-004			X														
RU-C-005				X													
RU-C-006														X			
RU-C-007					X												



RU-C-008																X	
RU-C-009						X											
RU-C-010																X	
RU-C-011								X									
RU-C-012																	X
RU-C-013									X								
RU-C-014							X										

Tabla 92. Matriz de trazabilidad Requisitos de Usuario – Casos de Uso.

5.7.1.2 Matrices de Requisitos software-Casos de uso

En esta matriz se observa la relación entre los casos de uso recogidos durante la fase de Estudio de Viabilidad del Sistema y los requisitos funcionales de software catalogados durante la fase de Análisis, quedando así demostrada la coherencia entre ambas fases.

	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013	CU-014	CU-015	CU-016	CU-017
RS-F-001	X																
RS-F-002	X																
RS-F-003	X																
RS-F-004	X																
RS-F-005	X																



RS-F-006		X															
RS-F-007										X							
RS-F-008									X								
RS-F-009												X					
RS-F-010											X						
RS-F-011			X														
RS-F-012				X													
RS-F-013													X				
RS-F-014													X				
RS-F-015													X				
RS-F-016					X												
RS-F-017														X			
RS-F-018						X											
RS-F-019																X	
RS-F-020							X										
RS-F-021																	X
RS-F-022								X									
RS-F-023							X										

Tabla 93. Matriz de trazabilidad de Requisitos software – Casos de Uso.



5.7.2 Validación de los modelos

Se ha comprobado la integridad de las matrices de trazabilidad observando que todos los casos de uso son recogidos ya sea en requisitos de usuario de capacidad o como requisitos software de funcionalidad. Esto produce que los modelos planteados en las distintas fases, en este caso Estudio de Viabilidad del Sistema y la fase de Análisis, sean consistentes y coherentes.



6. DISEÑO DEL SISTEMA

El diseño de sistema es el arte de definir la arquitectura de hardware y software, componentes, módulos y datos de un sistema para satisfacer ciertos requerimientos.

6.1 Introducción

El objetivo principal de este capítulo es presentar el diseño del sistema realizado de forma detallada así como estudiar la tecnología que será de utilidad para llevar a cabo esta actividad.

6.1.1 Propósito

Con este fin se realizará una especificación detallada con el fin de cubrir todas las decisiones de diseño correspondientes a la fase de construcción. De esta manera, sólo quedarán posibles decisiones a tomar relacionadas con el lenguaje de programación seleccionado para la codificación del sistema.

El capítulo de Diseño del Sistema es de vital importancia en el desarrollo de un proyecto software, ya que marca las pautas para las posteriores fases de construcción y de implantación final del sistema, siendo un documento básico para los programadores encargados de la implementación.

6.2 Alcance

El presente capítulo presenta una especificación detallada del proyecto, permitiendo cubrir todas las decisiones de diseño que se verán reflejadas en la fase de construcción.

Por otro lado, se especificará el entorno tecnológico necesario para que el sistema pueda ejecutarse y, además, contendrá la planificación de capacidades, los requisitos de administración, el control de accesos, la seguridad y la operación.

A continuación se describe con mayor detalle las principales tareas a llevar a cabo:

- **Definición de la arquitectura del sistema:** Se realizará una división del sistema con el objetivo de organizar y facilitar el diseño. Estos subsistemas serán partes lógicas y coherentes que tendrán interfaces de comunicación entre ellas bien definidas.
- **Diseño de la arquitectura de soporte:** En esta tarea se determinarán los mecanismos genéricos de diseño que servirán posteriormente de ayuda para el diseño detallado del sistema e información. Por tanto se deberá realizar en paralelo con el diseño detallado, ya que hay mucho en común entre ambas actividades y los cambios de una tarea afectan a la otra.



- **Díselo de casos de uso reales:** Se partirá de los casos de uso identificados en la fase de análisis del sistema y se relacionarán con las clases identificadas en el diseño en lugar de trabajar con el modelo conceptual del modelo de Análisis.
- **Diseño de clases:** El objetivo es transformar el modelo de clases del análisis en un modelo de clases de diseño. Dicho modelo es mucho más extenso que el anterior, ya que recoge información de atributos, métodos y las relaciones que existen entre las distintas clases. Para ello, es fundamental considerar las especificaciones tecnológicas elegidas para la implementación.
- **Diseño físico de datos:** Indica el formato de los ficheros que la aplicación va a utilizar.
- **Verificación y aceptación de la arquitectura del sistema:** En este apartado se comprobará la trazabilidad entre componente, clases y casos de uso.
- **Especificación técnica del plan de pruebas:** Se especificará en detalle el entorno de pruebas a realizar para comprobar el correcto funcionamiento del sistema.
- **Establecimiento de requisitos de implantación:** En esta actividad se completará el catálogo de requisitos con aquellos relacionados con la implantación del sistema en el entorno. Todo esto permite preparar los medios y recursos necesarios para que los usuarios puedan manejar la aplicación sin problemas.

Es importante destacar que la información que aquí se presenta es una adaptación del diseño planteado por *Métrica Versión 3*, ya que debido a las características del proyecto (se trata de un proyecto pequeño) algunas de las actividades son demasiado extensas para el problema tratado.



6.3 Definición de la arquitectura

La arquitectura del sistema a desarrollar en el presente proyecto no atiende a ninguno de los modelos estandarizados, utiliza una arquitectura específica en la que interactúan los componentes entre sí. No se realizará una identificación de subsistemas, ya que este proyecto carece de ellos, ni se realizará un estudio sobre los riesgos asociados al entorno tecnológico ya que estos son mínimos debido al software utilizado.

6.3.1 Definición de los niveles de la arquitectura

Básicamente, el sistema se compone de un único módulo que ha de comunicarse con un módulo externo formado por las bases de datos de *Facebook*, con las que tendría que estar conectado el sistema a través de internet para poder obtener los datos necesarios para la partida.

6.3.2 Especificación de estándares y normas de diseño y construcción

Entre las actividades a realizar en este proyecto se encuentra la codificación del sistema, que hace indispensable la tarea de definir un estándar que sirva de guía tanto a diseñadores como a programadores a la hora de generar código fuente. A continuación se detallan las consideraciones a tener en cuenta.

6.3.2.1 Nombres de Ficheros

Los nombres de los ficheros que conforman el código fuente del sistema deberán ser representativos de la funcionalidad que contiene cada uno. Además, su longitud no deberá superar los 25 caracteres y la primera letra deberá estar escrita en mayúscula. En caso de ser un nombre compuesto, la primera letra de cada palabra será escrita en mayúsculas y el resto en minúsculas. Por ejemplo *NombreDelFichero*.

Finalmente, los archivos que serán subidos a la red serán una serie de archivos contenedores creados automáticamente por la herramienta con la que será codificado el sistema, *Adobe Flash Builder 4*, por lo que los nombres de estos ficheros dependerán de dicha herramienta.

6.3.2.2 Idioma

El idioma predominante a lo largo de todo el desarrollo del proyecto será el castellano, por lo que los comentarios introducidos en el código fuente deberán realizarse en este idioma. En cambio, a la hora de codificar el sistema, los nombres de procedimientos, funciones, variables, etc. estarán en inglés, aprovechando que los métodos del lenguaje se encuentran en dicho idioma así como todos los métodos relacionados con las librerías que *Facebook* pone a disposición del programador.



6.3.2.3 Cabeceras de las clases

Cada una de las clases generadas mostrará al principio de su código un conjunto de líneas de comentario siguiendo el formato de la figura 89:

```
/* Nombre de la clase:  
   Descripción:  
   Autor:  
*/
```

Figura 89. Cabeceras de las clases.

6.3.2.4 Visibilidad de las clases

Las clases podrán ser públicas (public) o privadas (private).

6.3.2.5 Identificadores de las clases

Los identificadores de las clases estarán formados únicamente por letras. Como excepción, la letra “ñ” será representada mediante los caracteres “ny”.

Además, los identificadores siempre comenzarán con una letra mayúscula. En el caso de que sea un nombre compuesto, la primera letra a partir de la segunda palabra será la mayúscula también. Ejemplo: *IdentificadorCompuesto*

6.3.2.6 Constantes

En caso de utilizar constantes, sólo se utilizarán las letras minúsculas. En caso de ser una palabra compuesta, la primera letra de la primera palabra empezará por minúscula y las siguientes por mayúsculas. Ejemplo: *constanteVariasPalabras*

6.3.2.7 Variables

Las nombres de variables estarán formados por letras minúsculas. Al igual que con las constantes, en el caso de ser una palabra compuesta, la primera letra de cada palabra posterior a la primera será mayúscula. Ejemplo: *nombreVariable*

6.3.2.8 Funciones y procedimientos

En cuanto a las funciones y procedimientos, es muy importante que el nombre describa con la mayor precisión posible la funcionalidad que desempeña. Al igual que con las constantes y variables, los nombres deberán ser escritos en letras minúsculas utilizando las mayúsculas en la primera letra de cada palabra en caso de ser un nombre compuesto. Ejemplo: *nombreFunción*



6.3.3 Especificación del entorno tecnológico

En este apartado se describe el conjunto de especificaciones referentes al entorno tecnológico del sistema diseñado. Se definirán el conjunto de elementos que compondrán el eje central de la infraestructura técnica que va a adoptar el sistema, además de abordar aquellos inconvenientes técnicos que puedan afectar el desarrollo del proyecto.

6.3.3.1 Hardware

Para el desarrollo del proyecto se emplearán dos sistemas: un ordenador PC con microprocesador AMD Athlon x4 y un ordenador portátil Hewlett-Packard.

6.3.3.2 Software

El sistema operativo instalado en la máquina empleada para la ejecución del sistema será Windows XP.

6.3.3.3 Comunicaciones

El sistema deberá comunicarse con las bases de datos que pone *Facebook* a disposición de los desarrolladores. Para hacerlo se emplearán unas librerías proporcionadas por *Facebook* y que permitirán realizar consultas a estas bases de datos utilizando un lenguaje similar a SQL.

6.3.4 Especificación de requisitos de seguridad y operación

Para analizar la especificación de los requisitos de operación se remite al lector al apartado 5.3.1.3. Requisitos de operación.

6.3.5 Estudio de la seguridad requerida en el proceso de diseño del sistema

Al tratarse de un sistema que no maneja información sensible para el usuario del propio sistema no es necesario supervisar la seguridad de las actividades del proceso de diseño.

Por otro lado, con el objetivo de evitar posibles pérdidas de información se realizarán backups regularmente.

6.4 Diseño de la arquitectura de soporte

En este apartado se presentan los elementos más significativos del sistema: los componentes de la arquitectura.

El sistema que ocupa el presente proyecto está compuesto por un único módulo que dispone de conectividad con las bases de datos de *Facebook*, figura 90. A continuación se describirán los componentes y la interacción entre ellos para posteriormente analizar las clases que los componen.

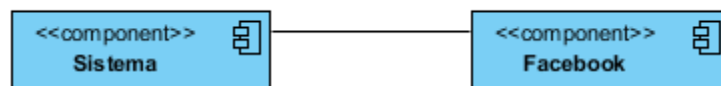


Figura 90. Componentes de la arquitectura del sistema

Con respecto a las características de los componentes que constituyen el sistema, éstas se presentarán en las tablas 94 y 95 indicando la información asociada a cada uno de los siguientes atributos:

- **Identificador:** Será único para cada uno de los componentes. Para ello se utilizará la nomenclatura *C-XXX*, donde *XXX* es el número del componente.
- **Nombre:** Nombre del elemento.
- **Tipo:** Tipo de elemento. En este caso será componente.
- **Propósito:** Fin del componente.
- **Función:** De qué se encarga el componente.
- **Subordinados:** Hace referencia a los elementos que de algún modo heredan algún dato del componente en cuestión.
- **Dependencias:** Componente que utiliza el elemento tratado para realizar sus funciones.
- **Interfaz:** Servicios proporcionados para el resto de elementos.
- **Recursos:** Hace referencia a elementos de tipo hardware que el componente puede necesitar para realizar su función.
- **Referencias:** Documentación necesaria para poder entender el componente.
- **Proceso:** Algoritmos especiales de procesamiento.
- **Datos:** Datos internos del elemento, atributos, relaciones de interés que no se hayan indicado en la descripción, o posibles valores de alguno de los elementos.

**Proyecto Fin de Carrera**

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

Identificador	C-001
Nombre	Sistema
Tipo	Componente
Propósito	Permite desarrollar partidas similares al popular juego de mesa <i>¿Quién es quién?</i> entre el usuario que accede a la aplicación y una IA.
Función	Solicita datos sobre el usuario conectado a la aplicación a las bases de datos de <i>Facebook</i> de manera que estos datos puedan ser utilizados durante el desarrollo del juego.
Subordinados	No aplica
Dependencias	<i>Facebook</i>
Interfaz	No aplica
Recursos	Conexión a Internet
Referencias	No aplica
Proceso	No aplica
Datos	No aplica

Tabla 94. Componente C-001: Sistema

Identificador	C-002
Nombre	<i>Facebook</i>
Tipo	Componente
Propósito	Componente externo que suministra datos e información al sistema acerca del usuario conectado a la aplicación.
Función	El sistema, mediante llamadas y consultas, solicita datos a <i>Facebook</i> acerca del usuario conectado y sus amigos, de modo que éstos puedan ser usados durante el desarrollo de la partida.
Subordinados	No aplica
Dependencias	No aplica
Interfaz	No aplica
Recursos	Conexión a Internet
Referencias	No aplica
Proceso	No aplica
Datos	No aplica

Tabla 95. Componente C-002: *Facebook*.



6.5 Diseño de casos de uso reales

En este apartado se estudiarán los casos de uso originados durante la fase de Análisis con el fin de identificar las clases de diseño necesarias para el sistema así como las relaciones entre ellas.

El objetivo será identificar el mínimo número de clases necesarias para el sistema siempre y cuando permitan cubrir todos los requisitos especificados anteriormente.

6.5.1 Identificación de clases asociadas a un caso de uso

En la tabla 96 se muestran las clases identificadas en el diseño y su asociación con los distintos casos de uso del sistema.

Identificador	Clases asociadas
CU-001	<ul style="list-style-type: none">• Aplicación• Facebook• Interfaz Usuario
CU-002	<ul style="list-style-type: none">• Interfaz Usuario• Interfaz Com
CU-003	<ul style="list-style-type: none">• Interfaz Usuario• Instrucciones
CU-004 CU-005 CU-006 CU-007 CU-008	<ul style="list-style-type: none">• Interfaz Usuario
CU-009	<ul style="list-style-type: none">• Interfaz Usuario• Interfaz Com• Aplicación
CU-010 CU-011	<ul style="list-style-type: none">• Interfaz Usuario
CU-012 CU-013	<ul style="list-style-type: none">• Interfaz Com
CU-014 CU-015	<ul style="list-style-type: none">• Interfaz Usuario

CU-016	<ul style="list-style-type: none"> Interfaz Com
CU-017	<ul style="list-style-type: none"> Interfaz Usuario

Tabla 96. Clases de diseño asociadas a casos de uso.

6.5.2 Diseño de la realización de los casos de uso

En este apartado se especifican de forma extendida los casos de uso detallados en el análisis haciendo corresponder los distintos escenarios con acciones concretas identificadas en las clases obtenidas en el desarrollo del documento.

- Caso de uso CU-001. Iniciar juego: Usuario iniciará la aplicación, figura 91, Aplicación cargará todos los datos necesarios para el desarrollo del juego, ya sea mediante consultas FQL o llamadas directamente a la API de *Facebook*. Una vez que aplicación haya terminado de cargar, Usuario podrá iniciar el juego, entonces Aplicación cargará la Interfaz Usuario dando comienzo a la partida.

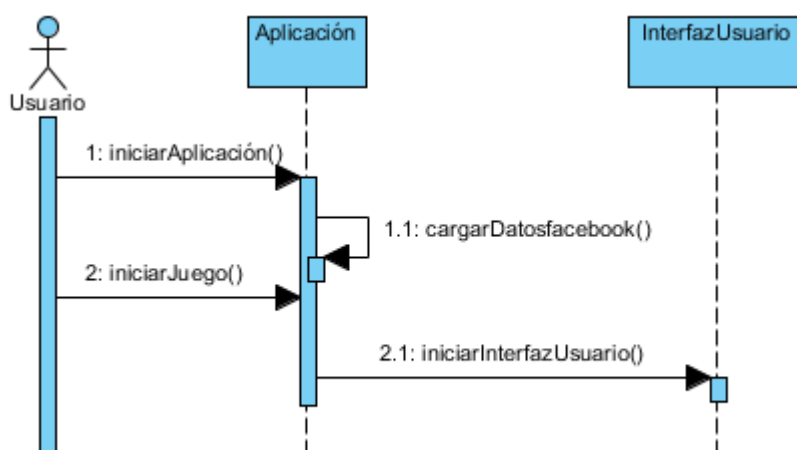


Figura 91. Diagrama de secuencia extendido. Caso de uso CU-001. Iniciar juego.

- Caso de uso CU-008. Resolver Usuario. Escenario Alternativo: Al comienzo de su turno, Usuario decide resolver, seleccionará del tablero presente en la Interfaz Usuario a un amigo para comprobar si este es la solución a descubrir. Usuario seleccionará resolver, la Interfaz Usuario comprobará si Usuario está en lo cierto, figura 92. En caso de no ser así, Interfaz Usuario volverá a cargar los datos necesarios para el tablero de Usuario, mediante consultas FQL o llamadas directamente a la API de *Facebook*. Interfaz Usuario hará saber a Usuario que no ha acertado y deberá empezar de nuevo. Finalmente, Usuario cederá su turno a Com.

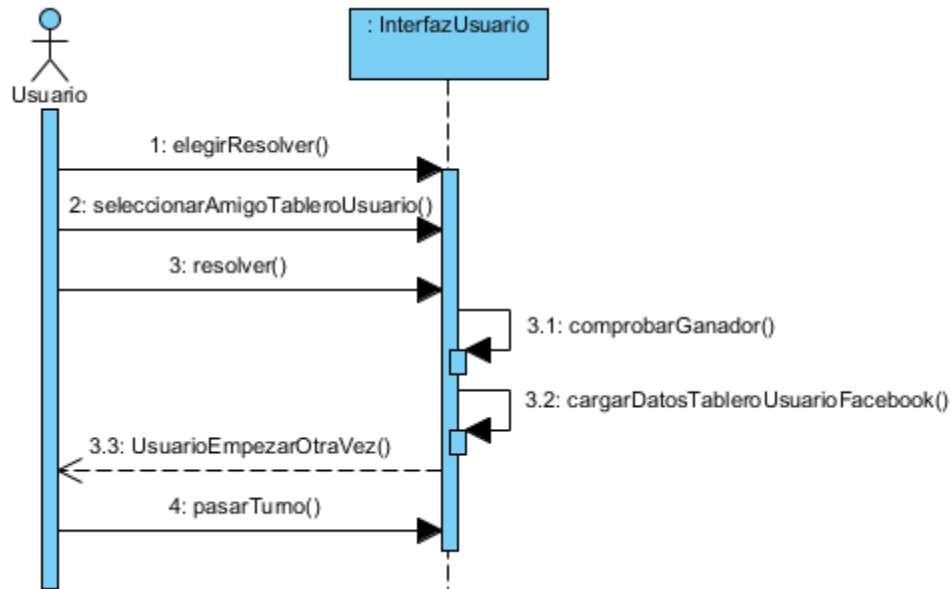


Figura 92. Diagrama de secuencia extendido. Caso de uso CU-008. Resolver Usuario. Escenario Alternativo.

6.6 Diseño de clases

En esta sección se desarrolla el modelo de clases del sistema que se está diseñando una vez completado el realizado durante la fase de Análisis del Sistema.

Los modelos aquí presentados servirán de guía al programador, por lo que los identificadores, relaciones, atributos y funciones son orientativos, dejando al programador libertad para programar bajo su propio criterio.

6.6.1 Identificación de clases de diseño

El modelo de clases de diseño del sistema resultante, y que se explicará en detalle en los siguientes subapartados, es el mostrado en la figura 93.

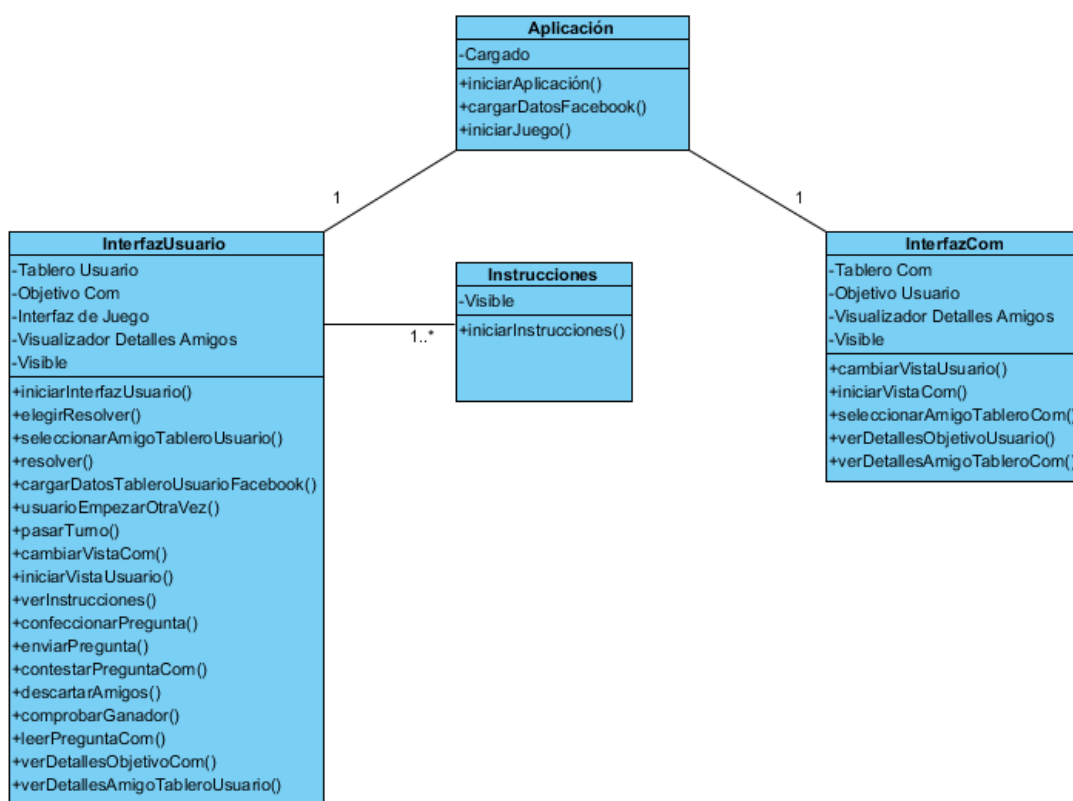


Figura 93. Diagrama de clases de diseño.



6.6.2 Identificación de atributos y métodos de clases de diseño

En las tablas 97-99 se describen las clases de diseño identificadas incluyendo el detalle de sus atributos y métodos.

CLASE	Aplicación
Descripción	Clase encargada de permitir al usuario conectarse a la aplicación y a <i>Facebook</i> para poder cargar los datos necesarios para la partida.
Dependencias	InterfazUsuario, InterfazCom
ATRIBUTOS	
Identificador	Cargado
Descripción	Indica cuando todos los datos necesarios han sido cargados de <i>Facebook</i> .
FUNCIONES Y PROCEDIMIENTOS	
Signatura	IniciarAplicación()
Descripción	Controla que el Usuario esté conectado a la red social <i>Facebook</i> y a la aplicación.
Signatura	CargarDatosFacebook()
Descripción	Realiza llamadas a las BB.DD. de <i>Facebook</i> para obtener todos los datos necesarios para el desarrollo del juego.
Signatura	IniciarJuego()
Descripción	Controla la funcionalidad asociada al botón <i>Empezar Juego</i> de la interfaz de la aplicación.

Tabla 97. Atributos, funciones y procedimientos clase Aplicación.



CLASE	InterfazUsuario
Descripción	Clase encargada de gestionar la interfaz de Usuario para que la aplicación pueda interactuar con éste.
Dependencias	Aplicación, Instrucciones.
ATRIBUTOS	
Identificador	TableroUsuario
Descripción	Tablero de juego de Usuario compuesto por 15 amigos/as elegidos al azar de su cuenta de <i>Facebook</i> .
Identificador	ObjetivoCom
Descripción	Objetivo a descubrir por Com elegido al azar de entre todos los amigos/as que forman el tablero de Com.
Identificador	InterfazDeJuego
Descripción	Controles que permiten el desarrollo del juego.
Identificador	VisualizarDetallesAmigo
Descripción	Permite observar más datos personales tanto de los amigos del tablero como del objetivo de Com.
Identificador	Visible
Descripción	Indica si esta interfaz está visible o oculta.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	iniciarInterfazUsuario()
Descripción	Aplicación hace visible el resto de la interfaz para que Usuario pueda jugar contra Com.
Signatura	elegirResolver()
Descripción	Controla la funcionalidad asociada al botón <i>Resolver</i> .
Signatura	SeleccionarAmigoTableroUsuario()
Descripción	Usuario podrá seleccionar un amigo del tablero para poder resolver u observar sus detalles.
Signatura	Resolver()
Descripción	Controla la funcionalidad del botón <i>Resolver</i> de la Etapa Resolver de la interfaz de Usuario.



Signatura	CargarDatosTableroUsuarioFacebook()
Descripción	La interfaz hará una llamada a las BB.DD. de <i>Facebook</i> para obtener datos que serán utilizados en el desarrollo de la partida.
Signatura	UsuarioEmpezarOtraVez()
Descripción	Controlará la funcionalidad asociada al botón <i>Falso, pierdes turno y empiezas de nuevo...</i> de la interfaz de Usuario en su etapa Empezar de Nuevo.
Signatura	PasarTurno()
Descripción	Controlará las funciones asociadas al botón <i>Turno</i> .
Signatura	CambiarVistaCom()
Descripción	Controlará la funcionalidad asociada al botón <i>Ir a vista Com</i> .
Signatura	IniciarVistaUsuario()
Descripción	La interfaz de Usuario se hará visible de nuevo cuando éste desee volver a la dicha interfaz.
Signatura	VerInstrucciones()
Descripción	Controlará la funcionalidad asociada al botón <i>Instrucciones</i> .
Signatura	ConfeccionarPregunta()
Descripción	Usuario podrá elegir una pregunta y la podrá completar en caso de que fuera necesario.
Signatura	EnviarPregunta()
Descripción	Controlará la funcionalidad asociada al botón <i>Preguntar</i> .
Signatura	ContestarPreguntaCom()
Descripción	Com responderá a la pregunta activando uno de los tres botones que representan las respuestas.
Signatura	DescartarAmigos()
Descripción	Controlará la funcionalidad asociada al botón <i>Descartar</i> .
Signatura	ComprobarGanador()
Descripción	Cuando Usuario decida resolver, se comprobará si ese era su objetivo a descubrir.
Signatura	LeerPreguntaCom()
Descripción	Usuario podrá observar la pregunta que le realiza Com en un cuadro de texto.



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

Signatura	VerDetallesObjetivoCom()
Descripción	Controlará la funcionalidad asociada al botón <i>Ver detalles del objetivo</i> .
Signatura	VerDetallesAmigoTableroUsado()
Descripción	Controlará la funcionalidad asociada al botón <i>Seleccione un amigo(a)...</i>

Tabla 98. Atributos, funciones y procedimientos clase Aplicación.



CLASE	InterfazCom
Descripción	Clase encargada de gestionar la interfaz de Com para que la aplicación pueda interactuar con Usuario.
Dependencias	Aplicación
ATRIBUTOS	
Identificador	TableroCom
Descripción	Tablero de juego de Com compuesto por 15 amigos/as elegidos al azar de la cuenta de <i>Facebook</i> de Usuario.
Identificador	ObjetivoUsuario.
Descripción	Objetivo a descubrir por Usuario elegido al azar de entre todos los amigos/as que forman el tablero de Usuario.
Identificador	VisualizarDetallesAmigo
Descripción	Permite observar más datos personales de los amigos que forman el tablero de Com.
Identificador	Visible
Descripción	Indica si esta interfaz está visible o oculta.
FUNCIONES Y PROCEDIMIENTOS	
Signatura	cambiarVistaUsuario()
Descripción	Controlará la funcionalidad asociada al botón <i>Ir a vista Player</i>
Signatura	iniciarVistaCom()
Descripción	Se ocultará la interfaz de Usuario haciendo visible la interfaz de Com.
Signatura	seleccionarAmigoTableroCom()
Descripción	Usuario podrá seleccionar un amigo del tablero de Com para poder observar sus detalles.
Signatura	VerDetallesObjetivoUsuario()
Descripción	Controlará la funcionalidad asociada al botón <i>Amigo/a objetivo. Pulse el botón para ver detalles.</i>
Signatura	VerDetallesAmigoTableroCom()
Descripción	Controlará la funcionalidad del botón <i>Seleccione un amigo/a para ver detalles.</i>

Tabla 99. Atributos, funciones y procedimientos clase Aplicación.

6.7 Diseño físico de los datos

Todos los datos empleados para el desarrollo de la partida están alojados en las bases de datos de *Facebook*. Éstas ponen a disposición de los desarrolladores una API sobre la que se pueden realizar una serie de peticiones de datos acerca de los amigos, eventos, imágenes, etc. Otra manera de obtener datos de las Bases de Datos de *Facebook* es mediante el uso del lenguaje *FQL*, parecido a *SQL*. De esta manera se pueden realizar consultas más complejas. Estas consultas se realizan sobre una serie de tablas que *Facebook* ha creado con información de los amigos, eventos, imágenes, etc. A continuación pasamos a describir la arquitectura de una aplicación Flash integrada en *Facebook*, figura 94. (Adobe Developer Connection, 2009)

1. El usuario solicita abrir la aplicación, mientras que en el sitio web de *Facebook*, el navegador hace una petición HTTP al servidor *Facebook*.
2. El servidor de *Facebook* devuelve una página HTML/JS que contiene el sitio web de *Facebook* y una etiqueta *iFrame* HTML.
3. El navegador del usuario hace una petición al servidor de la aplicación para mostrarla dentro del *iFrame*. Esta página incluirá un archivo SWF, la aplicación *Flash*.
4. El servidor devuelve una página HTML/JS en el navegador del usuario, mostrándola en el *iFrame*.
5. El navegador del usuario hace otra petición al servidor de la aplicación, para incrustar el archivo SWF en la página HTML que contiene el *iFrame*.
6. El servidor devuelve el archivo SWF.
7. A medida que usuario interactúa con la aplicación, hace llamadas a la API de *Facebook* usando el *Adobe ActionScript 3 SDK* para la red social.
8. *Facebook* devuelve datos en formato JSON a la aplicación con información de la conexión.
9. A medida que el usuario utiliza la aplicación, ésta puede realizar llamadas al servidor de la aplicación usando los métodos típicos de llamada a procedimiento remoto.
10. Si el servidor de la aplicación tiene que comunicarse con *Facebook*, lo realiza mediante una petición HTTP a la API.
11. El servidor también recibe datos de la conexión en formato JSON de *Facebook*.
12. El servidor devuelve datos a la aplicación del navegador del usuario.

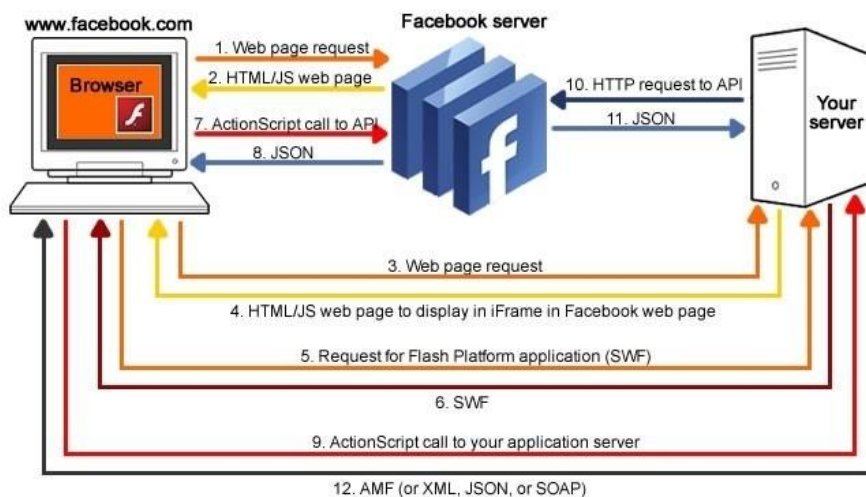


Figura 94. Arquitectura de una aplicación en *Facebook*.



6.8 Verificación y aceptación de la arquitectura del sistema

El objetivo de la tarea documentada en este apartado es el de garantizar la calidad y viabilidad de las especificaciones del diseño del sistema de información antes de realizar el diseño detallado.

6.8.1 Análisis de consistencia de las especificaciones de diseño

Esta tarea pretende asegurar que las especificaciones de diseño realizadas hasta el momento son coherentes, comprobando que no son ambiguas, que no hay información duplicada y que todos los casos de uso quedan cubiertos con el diseño. Para ello se traza esta matriz en la que las Clases de diseño recogen todos los Casos de uso obtenidos durante la fase de Estudio de Viabilidad del Sistema.

	CU-001	CU-002	CU-003	CU-004	CU-005	CU-006	CU-007	CU-008	CU-009	CU-010	CU-011	CU-012	CU-013	CU-014	CU-015	CU-016	CU-017
Aplicación	X								X								
InterfazUsuario	X	X	X	X	X	X	X	X	X	X	X			X	X		X
InterfazCom		X							X			X	X			X	
Instrucciones			X														

Tabla 100. Matriz de trazabilidad Clases de diseño – Casos de uso



6.9 Especificación Técnica del Plan de Pruebas

El objetivo de la tarea que se va a desempeñar en este apartado es el de especificar un plan detallado de las diferentes pruebas a realizar una vez implementado el sistema.

La elaboración de un plan de pruebas detallado tiene como finalidad comprobar que el sistema desarrollado cumple todas las funcionalidades requeridas, que no presenta errores y que proporciona un nivel de calidad suficiente.

El plan de pruebas se ha realizado tomando como punto de partida el catálogo de requisitos y el diseño detallado del sistema. Se puede analizar el plan de pruebas resultante en el apartado siete (Plan de Validación y Verificación del software).



6.10 Establecimiento de Requisitos de Implantación

En este apartado se especifican los requisitos de implantación del proyecto. Estos requisitos están relacionados con la formación, infraestructura e instalación, siendo necesarios para preparar y organizar la implantación del sistema de forma adecuada.

6.10.1 Especificación de Requisitos de Implantación

El identificador de cada uno de los requisitos de implantación tendrá el formato *RS-IMP-XXX*, donde *XXX* representará un valor numérico entre 000 y 999.

Una vez dicho esto, en la tabla 101 se recoge el único requisito de implantación existente en este proyecto.

Identificador	RS-IMP-001		
Prioridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Fuente	Equipo de desarrollo
Necesidad	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Verificabilidad	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción	Disponer de un ordenador con sistema operativo Microsoft Windows XP.		

Tabla 101. Requisito software RS-IMP-01



7. Plan de Validación y Verificación del Software

El objetivo de este capítulo es presentar la definición de un plan de pruebas que permita comprobar que el sistema funciona correctamente, cumpliendo con las especificaciones dadas por el cliente en su solicitud.

7.1 Introducción

A lo largo de este capítulo se identificarán los componentes de software y documentos que han de ser verificados. También se describirán las estrategias de verificación que serán utilizadas.

7.1.1 Propósito del documento

El plan de pruebas propuesto detalla todas las características a comprobar del sistema junto con los procedimientos a llevar a cabo para realizar dichas comprobaciones. Cada una de las pruebas a realizar debe llevar asociada una descripción completa, los pasos a seguir para su realización y los criterios para la aceptación de la prueba. Asimismo, se especifica también el entorno necesario para la realización de las pruebas, y el proceso de creación de informes de la pruebas realizadas.

Es destacable la importancia de una buena definición de requisitos software y hardware previa que permita garantizar una correcta evaluación del sistema, haciendo que los resultados de las pruebas realizadas se acerquen en la mayor medida posible a los obtenidos por el sistema en el entorno de explotación.

Teniendo como guía el presente capítulo se obtendrá un sistema verificado y validado funcionando correctamente y asegurando la satisfacción del cliente debido al cumplimiento de su solicitud.

7.1.2 Visión general del plan de Validación y Verificación del Sistema

El capítulo está dividido en los siguientes apartados:

- **Plan de prueba:** Esta sección describe todos y cada uno de los elementos del sistema que serán probados, así como las características que serán verificadas y los productos que deberán entregarse antes y después de la realización de las pruebas. El plan de prueba abarca también las tareas necesarias para preparar y llevar a cabo cada una de las pruebas, así como el entorno necesario para la ejecución de las mismas y el criterio de aceptación o rechazo que se va a aplicar a las pruebas.
- **Procedimientos de pruebas:** Recoge los procedimientos de prueba que se aplicarán sobre el sistema. Cada uno de ellos vendrá definido por un



identificador único, su descripción, las condiciones necesarias en el sistema, y los pasos a seguir para llevar a cabo la prueba.

- **Plantilla de informe de prueba:** Por cada prueba que se realice sobre el sistema se deberá rellenar un informe siguiendo el formato y el proceso que se describe en este apartado.
- **Matriz de Trazabilidad:** aunque la metodología ESA Lite no lo incluye para este documento, se realizará una matriz de trazabilidad entre requisitos de capacidad y pruebas con el fin de asegurar que todas las funcionalidades requeridas por el cliente serán probadas.



7.2 Plan de prueba

Es imprescindible establecer un plan de prueba para poder desarrollar un sistema informático de calidad y que funcione de manera correcta.

En este apartado se definen los objetivos de las pruebas a realizar y cómo han de llevarse a cabo. Debe quedar recogido qué elementos van a ser probados, las características del sistema que se van a identificar, los elementos entregables, las actividades a realizar para poder construir una prueba, qué requisitos debe tener el sistema para que las pruebas se ejecuten con éxito y, por último, los criterios a tener en cuenta por el equipo de desarrollo para determinar si una prueba es finalizada con éxito o no.

7.2.1 Elementos de prueba

La definición de pruebas que verifiquen si el producto funciona de manera correcta o si existen funcionalidades que no pasan los controles de calidad se hace imprescindible en un proyecto software. Para que la definición de las pruebas sea la correcta un paso previo e importante es establecer qué elementos del sistema van a someterse a prueba.

Todas las funcionalidades del sistema deben ser sometidas a pruebas, pero es necesario identificar cada una de las partes involucradas en el proceso de verificación y validación. De esta forma se puede controlar de forma más exhaustiva las pruebas de carácter que se han de realizar, tanto de elementos particulares, como generales.

De esta forma se identifican los siguientes elementos que serán sometidos a pruebas:

- **Aplicación:** Se realizarán pruebas que permitan comprobar una correcta conexión del usuario tanto con *Facebook* como con la aplicación, además de controlar que los datos obtenidos de las BB.DD. de *Facebook* son correctos y necesarios.
- **InterfazUsuario:** Se realizarán pruebas de todas sus funcionalidades, dándole una mayor importancia a las pruebas relacionadas con los controles de juego, de modo que se pueda desarrollar la partida con absoluta normalidad.
- **InterfazCom:** Se debe probar, principalmente, que el tablero haya sido cargado correctamente, así como el objetivo que el usuario ha de descubrir.
- **Instrucciones:** Se comprobará la correcta ejecución y visionado de las instrucciones mediante unas pruebas sencillas.



7.2.2 Características que se probarán

De los elementos de prueba identificados en el apartado anterior se probarán las características de estos. A continuación se detalla con más profundidad el conjunto de características a probar.

Para Aplicación:

- Conexión con *Facebook*
- Conexión con la Aplicación
- Carga de datos

Para InterfazUsuario:

- Cambiar vista a Com
- Salir
- Tablero Usuario
- Objetivo Com
- Detalles amigo/a tablero Usuario
- Detalles objetivo Com
- Realizar pregunta
- Respuesta Com
- Descartar
- Contestar pregunta Com
- Resolver
- Empezar de nuevo

Para InterfazCom:

- Cambiar Vista a Usuario
- Salir
- Tablero Com
- Objetivo Usuario
- Detalles amigo/a tablero Com
- Detalles objetivo Usuario

Para Instrucciones:

- Ejecutar Instrucciones
- Observar instrucciones



7.2.3 Entregables

Para poder probar cada uno de los elementos y funcionalidades descritas anteriormente, es necesario tener cierto conocimiento acerca de cómo funciona el sistema y de los recursos que son necesarios para el mismo, además de saber qué criterios se han definido para que el sistema pueda ser considerado válido y, por tanto, cumpla con las necesidades del cliente.

Para ello se hace indispensable disponer de una serie de documentos y recursos antes de la realización de pruebas. Éstos son:

- Documento de Estudio de Viabilidad.
- Documento de Análisis del sistema.
- Documento de Diseño del Sistema.
- Plan de Validación y Verificación del Software.
- El software del sistema.
- Manuales de usuario del sistema.

Una vez que se dispone de los recursos necesarios para la realización de las pruebas, se procederá a ejecutar cada una de ellas en el sistema. Los resultados de las pruebas serán recogidos en un informe denominado *Informe de realización de pruebas del sistema* en el que se registrará toda información acerca de las pruebas realizadas. Dicho informe deberá contener una tabla por cada prueba realizada siguiendo la plantilla definida para tal fin en el apartado 7.4 *Plantilla de informe de pruebas*.

7.2.4 Tareas de prueba

Desarrollar un sistema de calidad que cumpla con el resultado esperado por el cliente requiere definir un conjunto de pruebas lo más completo posible de forma que cada una de las funcionalidades del sistema se compruebe correctamente.

Debido a que el conjunto de pruebas puede llegar a ser excesivamente numeroso se hace necesaria la definición de un conjunto de tareas que sirvan de guía para preparar y desarrollar cada una de las pruebas.

Con este fin, las tareas definidas para desarrollar las pruebas de este proyecto son las siguientes:

- **Definición de la prueba de aceptación:** Se debe realizar una descripción acerca de la prueba especificando cual es el objetivo de la misma, qué elementos del sistema serán sometidos a prueba y las características que debe tener el sistema para que la prueba pueda ser ejecutada. Además se detallará cada uno de los pasos a realizar para la correcta ejecución de la prueba.
- **Ejecución de la prueba de aceptación:** Observando el comportamiento del sistema y los resultados obtenidos se podrá determinar si la prueba es superada con éxito o no.



- **Registro de la prueba de aceptación:** Deberá generarse una tabla con la información acerca de la prueba, los resultados y los posibles fallos encontrados. Esta tabla debe cumplir el formato definido en el apartado 7.4 *Plantilla de informe de pruebas*.

7.2.5 Necesidades del entorno

Para que las pruebas definidas puedan ser ejecutadas de manera satisfactoria y que produzcan los mismos resultados que en el entorno en el que el sistema real debe ser implantado, es necesario definir las características que debe tener el sistema donde se van a realizar las pruebas.

En el caso de este proyecto, el entorno de pruebas tendrá las mismas características que el entorno de implantación del sistema, por lo que en este apartado no es necesario definir nuevas necesidades para el entorno.

7.2.6 Criterio de aceptación/rechazo de cada caso de prueba

Al realizar la ejecución de cada uno de los casos de prueba definidos en el plan es necesario determinar si la prueba se considera válida, porque el sistema funciona como debería, o no, si presenta algún tipo de fallo en alguna de las funcionalidades requeridas.

El criterio establecido en este proyecto para considerar una prueba como *Aceptada*, es que dicha prueba se ejecute según lo definido, es decir, el sistema tendrá que responder de la manera exacta que detalle el caso de prueba correspondiente. En el caso de que el sistema produzca algún tipo de fallo o proporcione un resultado diferente al esperado, la prueba será *Rechazada*.



7.3 Procedimientos de prueba

En este apartado se especifican las pruebas que se deben realizar al sistema para comprobar su correcto funcionamiento y si cumple con todas las especificaciones.

Para ello, cada una de las pruebas explicadas en las tablas 102-122 incluirá los siguientes campos para su especificación:

- **Identificador:** Nombra de manera única a la prueba siguiendo el formato *PR-XXX*, donde *XXX* tomará valores numéricos dentro del rango 000-999.
- **Descripción:** Especifica el objetivo de la prueba.
- **Precondiciones:** Describe el estado en el que se debe encontrar el sistema antes de la realización de la prueba.
- **Pasos a seguir:** Establece un secuencia de pasos que debe realizar el responsable de la prueba sobre el sistema.

Identificador	PR-001
Descripción	Arrancar la aplicación correctamente desde <i>Facebook</i> .
Precondiciones	Ninguna.
Pasos a seguir	1. Insertar la dirección web que proporciona <i>Facebook</i> . 2. Comprobar que la aplicación arranca correctamente.

Tabla 102. Prueba PR-001

Identificador	PR-002
Descripción	Arrancar la aplicación correctamente desde la página web.
Precondiciones	Ninguna.
Pasos a seguir	1. Insertar la dirección web que proporciona nuestro servidor. 2. Comprobar que la aplicación arranca correctamente.

Tabla 103. Prueba PR-002



Identificador	PR-003
Descripción	Cargar datos de <i>Facebook</i> .
Precondiciones	Aplicación iniciada
Pasos a seguir	<ol style="list-style-type: none">1. Aplicación comprueba los permisos de Usuario para ver si puede acceder a los datos necesarios2. Aplicación realiza consultas a las BB.DD. de <i>Facebook</i>.3. Comprobar que los datos recibidos sean los deseados.

Tabla 104. Prueba PR-003

Identificador	PR-004
Descripción	Ver tablero de Com.
Precondiciones	Datos cargados.
Pasos a seguir	<ol style="list-style-type: none">1. Usuario solicita ver el tablero de Com.2. Comprobar que la aplicación permite ver correctamente el tablero de Com y su vista.

Tabla 105. Prueba PR-004

Identificador	PR-005
Descripción	Observar detalles del Objetivo de Usuario.
Precondiciones	Juego iniciado, y Usuario se encuentra en interfazUsuario.
Pasos a seguir	<ol style="list-style-type: none">1. Usuario hace clic sobre <i>Ver detalles del objetivo</i>.2. InterfazUsuario muestra los detalles de éste.3. Comprobar si son correctos.

Tabla 106. Prueba PR-005



Identificador	PR-006
Descripción	Observar detalles del Objetivo de Com.
Precondiciones	Juego iniciado, y Usuario se encuentra en interfazCom.
Pasos a seguir	<ol style="list-style-type: none">1. Usuario hace clic sobre <i>Amigo/a objetivo</i>. Pulse sobre el botón para ver detalles.2. InterfazCom muestra los detalles de éste.3. Comprobar si son correctos.

Tabla 107. Prueba PR-006

Identificador	PR-007
Descripción	Observar detalles de un amigo del tablero Usuario.
Precondiciones	Juego iniciado, y Usuario se encuentra en interfazUsuario.
Pasos a seguir	<ol style="list-style-type: none">1. Usuario selecciona un amigo de entre todos los que aparecen en el tablero de Usuario.2. Usuario hace clic sobre el botón <i>Ver detalles + nombre</i>.3. InterfazUsuario muestra los datos del amigo en cuestión.4. Comprobar que los datos son correctos.

Tabla 108. Prueba PR-007



Identificador	PR-008
Descripción	Observar detalles de un amigo del tablero Com.
Precondiciones	Juego iniciado, y Usuario se encuentra en interfazCom.
Pasos a seguir	<ol style="list-style-type: none">1. Usuario selecciona un amigo de entre todos los que aparecen en el tablero de Com.2. Usuario hace clic sobre el botón <i>Ver detalles + nombre</i>.3. InterfazCom muestra los datos del amigo en cuestión.4. Comprobar que los datos son correctos.

Tabla 109. Prueba PR-008

Identificador	PR-009
Descripción	Consultar página web de instrucciones.
Precondiciones	Juego iniciado, y Usuario se encuentra en interfazUsuario.
Pasos a seguir	<ol style="list-style-type: none">1. Usuario hace clic sobre el botón <i>Instrucciones</i> de la interfazUsuario.2. Se abre una nueva ventana del navegador con la página web que contiene las instrucciones y reglas del juego.3. Comprobar que el contenido de la página web es el esperado .

Tabla 110. Prueba PR-009



Identificador	PR-010
Descripción	Usuario Realiza Pregunta no completar X , donde X se corresponde con la numeración de la pregunta. Las posibles preguntas son 1, 2, 6, 7, 8, 9, 10, 11, 12, 13, 14.
Precondiciones	Juego iniciado, y Usuario se encuentra en interfazUsuario en el principio de su turno.
Pasos a seguir	<ol style="list-style-type: none">1. Usuario selecciona la pregunta número X de entre todas las disponibles.2. Usuario hace clic sobre el botón <i>Enviar</i>, para mandar la pregunta a Com.3. Comprobar que la pregunta ha sido bien formulada.

Tabla 111. Prueba PR-010

Identificador	PR-011
Descripción	Usuario Realiza Pregunta completar X , donde X se corresponde con la numeración de la pregunta. Las posibles preguntas son 3, 4, 5, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26.
Precondiciones	Juego iniciado, y Usuario se encuentra en interfazUsuario en el principio de su turno.
Pasos a seguir	<ol style="list-style-type: none">1. Usuario selecciona la pregunta número X de entre todas las disponibles.2. Usuario completa la pregunta rellenando el recuadro de texto dedicado para ello con información.3. Usuario hace clic sobre el botón <i>Enviar</i>, para mandar la pregunta a Com.4. Comprobar que la pregunta ha sido bien formulada.

Tabla 112. Prueba PR-011

Identificador	PR-012
Descripción	Com Realiza Pregunta no completar X , donde X se corresponde con la numeración de la pregunta. Las posibles preguntas son 1, 2, 6, 7, 8, 9, 10, 11, 12, 13, 14.
Precondiciones	Juego iniciado, Usuario se encuentra en interfazUsuario, y es turno de Com.
Pasos a seguir	<ol style="list-style-type: none">1. Com hará la pregunta X a Usuario en caso de no haberla realizado antes.2. Com enviará la pregunta, haciéndose visible ésta en la interfazUsuario.3. Comprobar que la pregunta ha sido bien formulada.

Tabla 113. Prueba PR-012



Identificador	PR-013
Descripción	Usuario Realiza Pregunta completar X , donde X se corresponde con la numeración de la pregunta. Las posibles preguntas son 3, 4, 5, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26.
Precondiciones	Juego iniciado, Usuario se encuentra en interfazUsuario, y es turno de Com.
Pasos a seguir	<ol style="list-style-type: none">1. Com hará la pregunta X a Usuario en caso de no haberla realizado antes.2. Com completa la pregunta con la información que pertenece a un amigo de su tablero que todavía no se encuentre descartado.3. Com enviará la pregunta haciéndose visible en la interfazUsuario.4. Comprobar que la pregunta ha sido bien formulada.

Tabla 114. Prueba PR-013

Identificador	PR-014
Descripción	Respuesta Usuario X , donde X corresponde a cada una de las posibles respuestas que Usuario puede dar: Si, No, Sin detalles.
Precondiciones	Juego iniciado, Usuario se encuentra en interfazCom, es turno de Com.
Pasos a seguir	<ol style="list-style-type: none">1. Usuario lee la pregunta lanzada por Com.2. Usuario hace clic sobre el botón X para contestar a la pregunta.3. Comprobar que la respuesta llega perfectamente a Com.

Tabla 115. Prueba PR-014

Identificador	PR-015
Descripción	Respuesta Com X , donde X corresponde a cada una de las posibles respuestas que Usuario puede dar: Si, No, Sin detalles.
Precondiciones	Juego iniciado, Usuario se encuentra en interfazUsuario, es turno de Usuario.
Pasos a seguir	<ol style="list-style-type: none">1. Com responderá X a la pregunta lanzada por Usuario.2. Se hará visible la respuesta de Com en la interfazUsuario activándose el botón X.3. Comprobar que Com ha dado la respuesta correcta.

Tabla 116. Prueba PR-015



Identificador	PR-016
Descripción	Descartar amigos/as tablero Usuario.
Precondiciones	Juego iniciado, Usuario se encuentra en interfazUsuario, es turno de Usuario y éste podrá observar la respuesta de Com a la pregunta previamente realizada.
Pasos a seguir	<ol style="list-style-type: none">1. Usuario en función de la respuesta de Com, seleccionará del tablero aquellos que no cumplan la condición.2. Usuario hará clic sobre el botón <i>Descartar</i>.3. InterfazUsuario mostrará qué amigos han sido descartados del tablero.4. Comprobar que InterfazUsuario es capaz de mostrar correctamente que amigos han sido descartados.

Tabla 117. Prueba PR-016

Identificador	PR-017
Descripción	Descartar amigos/as tablero Com.
Precondiciones	Juego iniciado, es turno de Com y Usuario acaba de responder a la pregunta formulada por Com.
Pasos a seguir	<ol style="list-style-type: none">1. Com repasará todos los amigos del tablero que no hayan sido descartados comprobando que cumplen con la respuesta para no ser descartados.2. En la interfazCom se hará visible que amigos han sido descartados.3. Com cederá el turno a Usuario una vez que haya terminado de descartar.4. Comprobar que Com ha descartado correctamente y que InterfazCom lo muestra.

Tabla 118. Prueba PR-017



Identificador	PR-018
Descripción	Resolver Usuario.
Precondiciones	Juego iniciado, Usuario se encuentra en interfazUsuario, y es principio de turno para Usuario.
Pasos a seguir	<ol style="list-style-type: none">1. Usuario hará clic sobre el botón <i>Resolver</i>.2. Usuario seleccionará de su tablero a un amigo.3. Usuario hace clic de nuevo sobre el botón <i>Resolver</i>.4. La interfazUsuario mostrará si es ganador o si por el contrario ha de empezar de nuevo.5. Comprobar que la solución es correcta.

Tabla 119. Prueba PR-018

Identificador	PR-019
Descripción	Resolver Com.
Precondiciones	Juego iniciado, Usuario se encuentra en interfazUsuario, y es turno de Com.
Pasos a seguir	<ol style="list-style-type: none">1. Al principio de su turno, Com comprobará cuantos Amigos le quedan sin descartar.2. En caso de quedar sólo uno resolverá proponiendo a éste como solución.3. Comprobar que Com ha resuelto de manera correcta.

Tabla 120. Prueba PR-019

Identificador	PR-020
Descripción	Salir del juego.
Precondiciones	Ninguna.
Pasos a seguir	<ol style="list-style-type: none">1. Usuario hará clic sobre el botón <i>Salir</i>.2. La aplicación parará su ejecución, además de desconectar a Usuario de <i>Facebook</i>.3. Comprobar que Usuario está desconectado de <i>Facebook</i> y la aplicación ha parado su ejecución

Tabla 121. Prueba PR-020



Identificador	PR-021
Descripción	Usuario ceder turno.
Precondiciones	Juego iniciado, Usuario se encuentra en interfaz Usuario, es turno de Usuario y éste habrá terminado de descartar.
Pasos a seguir	<ol style="list-style-type: none">1. Una vez que haya descartado, Usuario hará clic sobre el botón <i>Turno</i>.2. Le será cedido el turno a Com.3. Comprobar que el turno le ha sido cedido correctamente a Com.

Tabla 122. Prueba PR-021



7.4 Plantilla de informe de pruebas

Durante la realización de las pruebas del sistema se debe realizar un informe en el que se detallen los resultados obtenidos. A continuación se especifican los campos que deben ser incluidos en el informe y el formato establecido para este proyecto en la figura 123:

- **Identificador:** Identificador de la prueba realizada.
- **Fecha:** Fecha en la que se realizó la prueba.
- **Responsable:** Miembro del equipo del proyecto encargado de realizar la prueba.
- **Descripción:** Objetivo de la prueba realizada.
- **Resultado:** Este campo mostrará el valor *Aceptada* si el sistema ha realizado la prueba correctamente, o *Rechazada* en caso contrario.
- **Descripción de los fallos:** Si la prueba ha sido rechazada este campo contendrá una descripción de los fallos encontrados durante la realización de dicha prueba.

Identificador	
Fecha	
Responsable	
Descripción	
Resultado	
Descripción de los fallos	

Tabla 123. Plantilla informe de pruebas.



7.5 Matriz de trazabilidad Pruebas/Requisitos de Capacidad

En esta matriz se prueba la relación entre los requisitos de usuario de capacidad y las pruebas que se van a realizar, así se puede comprobar que todos los requisitos quedan cubiertos por pruebas.

	RU-C-001	RU-C-002	RU-C-003	RU-C-004	RU-C-005	RU-C-006	RU-C-007	RU-C-008	RU-C-009	RU-C-010	RU-C-011	RU-C-012	RU-C-013	RU-C-014
PR-001	X													
PR-002	X													
PR-003	X													
PR-004		X												
PR-005			X											
PR-006			X											
PR-007			X											
PR-008			X											
PR-009				X										
PR-010					X									
PR-011					X									
PR-012						X								
PR-013						X								
PR-014							X							
PR-015								X						
PR-016									X					
PR-017										X				
PR-018											X			



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

PR-019												X		
PR-020													X	
PR-021														X

Tabla 124. Matriz de trazabilidad Pruebas/Requisitos de capacidad.

8. Implementación del Sistema

Durante la fase de implementación se ha generado el código necesario para el funcionamiento de la RIA. Se han utilizado los lenguajes de programación *MXML* y *ActionScript* para desarrollar dicho código utilizando el entorno de desarrollo de *Adobe Flash Builder 4*.

En este apartado se describe el funcionamiento de las distintas interfaces de las que se compone el sistema además de un estudio minucioso de la implementación de la IA que controlará al actor Com.

8.1 Interfaz Aplicación

Esta es la primera interfaz con la que se encontrará el usuario de la aplicación, figura 95. Será esta interfaz la que permita a éste conectarse tanto con *Facebook* como con la aplicación.

Primero comprobará que la clave API es correcta, por lo que permitirá a la aplicación poder acceder a todos los métodos que pone a disposición *Facebook* para los desarrolladores.

Detectará si el usuario está intentando conectarse a la aplicación a través de *Facebook*, por lo que la aplicación estará contenida en un *iFrame* de dicha red social, o si por el contrario se está intentando acceder directamente desde una web independiente. En ambos casos, la interfaz Aplicación, comprobará si el usuario se ha logueado con su cuenta de *Facebook*, de modo que en caso de no ser así le redirigirá a páginas de *Facebook* para permitir el logueo.



Figura 95. Interfaz Aplicación funcionando bajo el *iFrame* de *Facebook*.

Una vez realizado el logueo, interfaz Aplicación comprobará si este usuario tiene aceptados los permisos necesarios para obtener los datos de los amigos que serán utilizados en el desarrollo del juego. En caso de no ser así, *Facebook* mostrará los permisos requeridos al usuario y le pedirá que acepte los términos.

Cuando usuario tenga los permisos aceptados, la interfaz Aplicación, comenzará a realizar las consultas a las bases de datos de *Facebook* utilizando tanto la API como las consultas *FQL* para la obtención de los datos referidos a los amigos del usuario.

Una vez que interfaz Aplicación tenga todos los datos necesarios para el desarrollo del juego, activará el botón *Empezar juego*, de modo que cuando el usuario haga clic sobre dicho botón dará comienzo el juego y se pasará a la siguiente interfaz.

8.2 Interfaz Usuario

Una vez que el usuario haya dado comienzo al juego se harán visibles todos los elementos de la interfaz Usuario y todos los controles que permitirán a este desarrollar la partida, figura 96.

Cada botón de esta interfaz tendrá un método asociado el cuál desempeñará una función que permita avanzar en el juego.



Figura 96. Interfaz Usuario funcionando bajo el iFrame de Facebook.

Tanto el diseño como los elementos que componen las interfaces están realizadas con *MXML*, por lo que a cada elemento le puede ser asociado un método o función, el cual ha sido desarrollado en *ActionScript*.

Esta interfaz está diferenciada en las distintas zonas y etapas que se mencionaron en el apartado 5.7.3.2 *Interfaz Usuario*. La interfaz está enfocada a permitir a Usuario poder desarrollar la partida de manera fluida y a que los controles sean lo más intuitivos posibles.

Permitirá la comunicación entre el usuario y el oponente (Com, IA desarrollada para competir) mediante un sistema de pregunta-respuesta que permita tanto a usuario como a Com ir descartando amigos de sus tableros de modo que puedan llegar a su objetivo.

8.3 Interfaz Com

Esta interfaz, figura 97, es accesible desde la interfaz Usuario. De manera que si usuario hace clic sobre el botón *Ir a vista Com* se ocultará toda la interfaz Usuario haciéndose visible la interfaz Com. En esta interfaz, Usuario podrá consultar el tablero de Com, de manera que puede ver los avances que éste va realizando con el descarte de amigos.



Figura 97. Interfaz Com funcionando bajo el iFrame de Facebook.

En esta vista también se podrán consultar los detalles de los amigos que forman el tablero de Com así como el objetivo que tiene que descubrir el usuario. Esta interfaz se puede entender como una interfaz de consulta.

Siempre se podrá volver a la interfaz Usuario haciendo clic sobre el botón *Ir a vista Player*.

8.4 Instrucciones

Se podrá acceder a las instrucciones y reglas del juego utilizando el botón *Instrucciones* que se encuentra en la interfaz Usuario. Al hacer esto se abrirá una ventana independiente del navegador rediriéndola a la web que contiene dichas normas, figura 98.



Figura 98. Interfaz Instrucciones.



8.5 IA de Com

Usuario se enfrentará a un inteligencia artificial (IA) que estará implementada en el sistema. En este apartado se describirán los algoritmos que determinarán el comportamiento de Com.

8.5.1 Responder

La primera función que ha de cumplir la IA será la de responder de manera correcta a las preguntas que le pueda formular el usuario. De este modo se puede desarrollar una partida sin sobresaltos. Para esta tarea, la IA, en función de la pregunta lanzada por el usuario, comprobará el campo del amigo-objetivo de Usuario al que hace referencia la pregunta dando su respuesta.

Por ejemplo, si el usuario pregunta *01. ¿Es hombre?*, la IA irá al campo *sexo* del amigo objetivo, de manera que se podrán producir tres respuestas distintas:

- Si: En este caso el amigo-objetivo es hombre.
- No: Se trata de una mujer.
- Sin detalles: El amigo-objetivo no tiene especificado su sexo.

En la interfaz Usuario se activará el botón correspondiente a la respuesta dada.

8.5.2 Preguntar

Otra de las tareas que debe realizar la IA será la de confeccionar preguntas. Para que la partida sea lo más competida posible, y en consecuencia más divertida, se hace imprescindible que Com realice sus preguntas de la manera más acertada posible. Por este motivo, la IA nunca repetirá una pregunta. Para esto, irá almacenando en un Array todas las preguntas realizadas.

La forma de actuar de IA a la hora de hacer una pregunta será la siguiente:

1. IA obtendrá un número al azar comprendido entre 0 y el número de preguntas totales, en este caso 26.
2. IA comprobará que dicha pregunta no ha sido realizada con anterioridad buscándola en el Array donde almacena las preguntas ya realizadas.
3. IA seleccionará un amigo al azar de su tablero. Esto lo hace por si la pregunta que tiene que formular ha de ser completada con algún campo. Por ejemplo, si la pregunta es del tipo *05. ¿Nació en el país de ...?*, necesita el campo *País* de algún amigo de su tablero para completar la pregunta. La única condición que tiene que tener el amigo en cuestión, es que no haya sido descartado previamente en algún turno anterior. Si esto se produjese, simplemente se repetiría este punto hasta dar con un amigo que no haya sido descartado.



4. IA esperará a la respuesta del usuario, y en función de la respuesta de éste se producirá el almacenamiento, o no, en el array de la pregunta realizada.

Para que la partida sea aún más competitiva, se ha intentado mejorar algunos aspectos a la hora de almacenar preguntas ya formuladas en el array. Por ejemplo, si Com pregunta 03. *¿Su ciudad de nacimiento es ...?* y el usuario responde que *Sin detalles*, esto quiere decir que el amigo-objetivo a descubrir por Com no tiene especificado su lugar de nacimiento. Por lo que si se formulan las preguntas 04. *¿Nació en la comunidad de ...?* o 05. *¿Nació en el país de ... ?* también responderá *Sin detalles*. Por lo que en estos casos, en los que la respuesta implique más preguntas, serán almacenadas todas las preguntas implicadas en el array.

8.5.3 Descartar

IA irá recorriendo todos los amigos que conforman su tablero e irá descartando a todos aquellos que no cumplan la respuesta que ha dado el usuario a la pregunta previamente formulada.

8.5.4 Resolver

Al principio de su turno, y antes de formular la pregunta, IA comprobará la cantidad de amigos que le quedan por descartar. En caso de ser sólo uno decidirá resolver ganando así la partida. Por el contrario, si queda más de uno, realizará otra pregunta.



9. Experimentación y Resultados

En este capítulo se presentan varias pruebas y experimentos realizados sobre el sistema ya desarrollado, mostrando así el funcionamiento del mismo. Para la realización de estas pruebas se van a seguir dos caminos. El primero consiste en una ejecución completa de la aplicación comprobando que cumple todos los requisitos del sistema recogidos. La segunda vía consiste en la realización de un test a rellenar por una serie de usuarios que utilicen la aplicación de modo que éstos puedan realizar una valoración de la jugabilidad, entorno, sencillez, etc. Además, de estos usuarios se recogerán datos como la edad, sexo y experiencia en aplicaciones para poder sacar mejores conclusiones.

9.1 Verificación de requisitos. Informe de Pruebas.

En este apartado se realizará una ejecución de la aplicación de manera que se pueda comprobar que se cumplen todos los requisitos de usuario, concluyendo que se ha hecho un software de calidad. Se llevarán a cabo los procedimientos de prueba recolectados anteriormente, de manera que de los resultados de la unión de la *matriz de trazabilidad de pruebas/Requisitos de Capacidad* y de la *matriz de requisitos de usuario de capacidad/Casos de uso* se sabrá si se han cumplido los objetivos marcados en los casos de uso recogidos durante el estudio de Viabilidad del Sistema.

9.1.1 Informe de pruebas.

Se utilizará la plantilla mostrada en el apartado 7.4 *Plantilla de informe de pruebas* para realizar los procedimientos de prueba definidos anteriormente, apartado 7.3 *Procedimientos de prueba*.

Identificador	PR-001
Fecha	28 de Septiembre de 2011
Responsable	Programador
Descripción	Arrancar la aplicación correctamente desde <i>Facebook</i> .
Resultado	Se inserta en el navegador la dirección correspondiente a la aplicación, figura 99, para funcionar bajo el <i>iFrame</i> de <i>Facebook</i> (https://apps.facebook.com/pfc-quienes/). Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 125. Plantilla prueba PR-001.



Figura 99. Realización de la prueba PR-001.

Identificador	PR-002
Fecha	28 de Septiembre de 2011
Responsable	Programador
Descripción	Arrancar la aplicación correctamente desde la página web.
Resultado	Se inserta en el navegador la dirección <i>http://pfc.comuf.com/</i> para acceder a la aplicación sin necesidad de estar bajo el <i>iFrame</i> de <i>Facebook</i> , figura 100. Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 126. Plantilla prueba PR-002

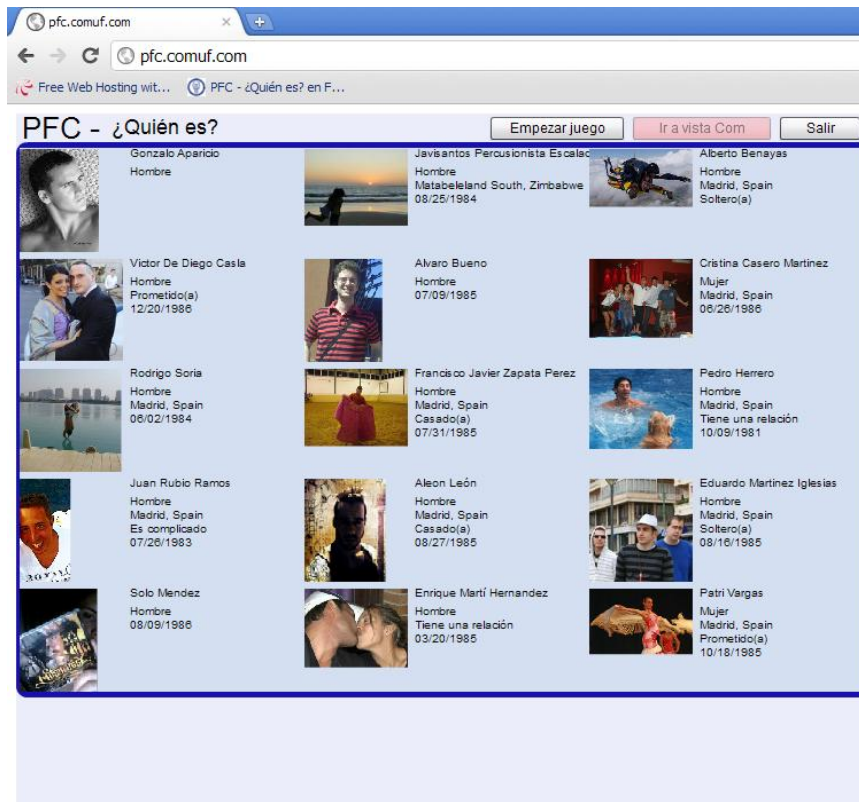


Figura 100. Realización de la prueba PR-002.

Identificador	PR-003
Fecha	29 de Septiembre de 2011
Responsable	Programador
Descripción	Cargar datos de <i>Facebook</i> .
Resultado	Se comprueba que se aceptan los permisos necesarios para poder acceder a los datos que ofrece <i>Facebook</i> , figura 101, y se comprueba que los datos recibidos son correctos mediante una comparación con los perfiles de los amigos. Prueba Superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 127. Plantilla prueba PR-003.

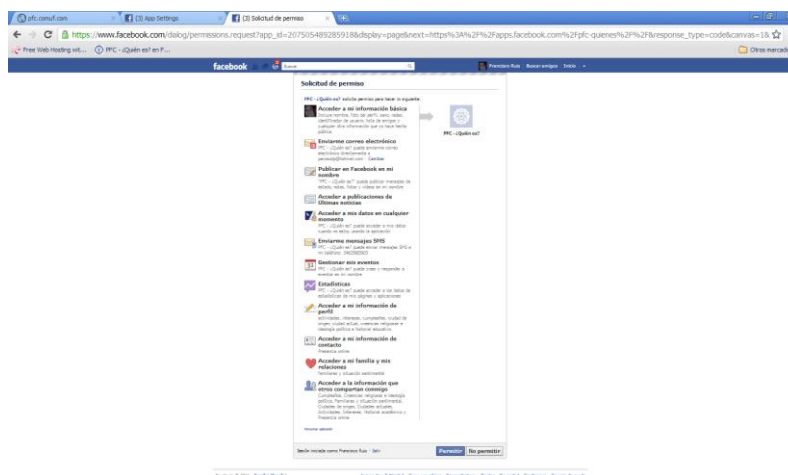


Figura 101. Realización de la prueba PR-003.

Identificador	PR-004
Fecha	29 de Septiembre de 2011
Responsable	Programador
Descripción	Ver tablero de Com.
Resultado	Se comprueba la funcionalidad del botón <i>Ir a vista Com.</i> figura 102. La aplicación muestra inmediatamente la vista de Com. Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 128. Plantilla prueba PR-004.



Figura 102. Realización de la prueba PR-004.

Identificador	PR-005
Fecha	29 de Septiembre de 2011
Responsable	Programador
Descripción	Observar detalles Objetivo Usuario.
Resultado	Se comprueba la funcionalidad del botón <i>Ver detalles del objetivo</i> , figura 103. Se comprueba que los datos mostrados son los deseados mediante una comparación con los del perfil del amigo. Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 129. Plantilla prueba PR-005.

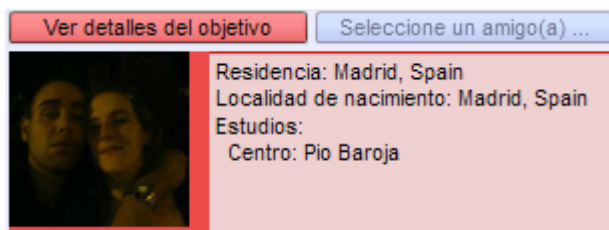



Figura 103. Realización de la prueba PR-005.



Identificador	PR-006
Fecha	30 de Septiembre de 2011
Responsable	Programador
Descripción	Observar detalles objetivo Com.
Resultado	Se comprueba la funcionalidad del botón <i>Amigo/a objetivo</i> , figura 104. <i>Pulse sobre el botón para ver detalles</i> de la <i>interfazCom</i> . Se comprueba que los datos aparecidos son los correctos comparándolos con los del perfil del amigo en <i>Facebook</i> . Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 130. Plantilla prueba PR-006.

Amigo/a objetivo. Pulse el botón para ver detalles.



Alvaro Bueno

Sexo: Hombre

Fecha de nacimiento : 07/09/1985

Seleccione un amigo(a) para ver detalles

Estudios:

Centro: Sagrada Familia- Jorge Juan

Centro: escuela politecnica alcalá

Centro: escuela politecnica alcalá

Figura 104. Realización de la prueba PR-006.



Identificador	PR-007
Fecha	30 de Septiembre de 2011
Responsable	Programador
Descripción	Observar detalles amigo tablero Usuario.
Resultado	Se comprueba la funcionalidad del botón <i>Ver detalles + nombre</i> de la interfaz Usuario, figura 105. Para ello se revisará que los datos aparecidos son los correctos comparándolos con los del perfil del amigo en <i>Facebook</i> . Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 131. Plantilla prueba PR-007.

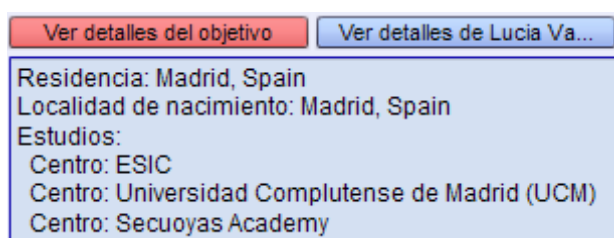


Figura 105. Realización de la prueba PR-007.

Identificador	PR-008
Fecha	30 de Septiembre de 2011
Responsable	Programador
Descripción	Observar detalles amigo tablero Com.
Resultado	Se comprueba la funcionalidad del botón <i>Ver detalles + nombre</i> de la interfaz Com, figura 106. Se revisará que los datos aparecidos son correctos. Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 132. Plantilla prueba PR-008.



Figura 106. Realización de la prueba PR-008.

Identificador	PR-009
Fecha	3 de Septiembre de 2011
Responsable	Programador
Descripción	Consultar página web de instrucciones.
Resultado	Se comprobará la funcionalidad del botón <i>Instrucciones</i> , figura 107. Para ello se revisará que la web abierta sea la correcta. Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 133. Plantilla prueba PR-009.



Figura 107. Realización de la prueba PR-009.




Identificador	PR-010
Fecha	3 de Septiembre de 2011
Responsable	Programador
Descripción	Usuario realiza pregunta no completar X.
Resultado	Se trata de una prueba genérica que deberá ser repetida para las preguntas indicadas en el proceso de prueba correspondiente. En el caso de la figura 108 se formula la pregunta 1 y se comprueba que la pregunta está bien formulada observando la respuesta que da Com a ésta. Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 134. Plantilla prueba PR-010.

Ver detalles del objetivo

Seleccione un amigo(a) ...



Alvaro Mejia Sevilla
Estado sentimental: Tiene una relación
Sexo: Hombre
Lugar de nacimiento: Madrid, Spain
Fecha de nacimiento : 03/28/1986

Seleccione y descarte a los amigos que no cumplan la respuesta.

01. ¿Es hombre?

Turno

Preguntar

RESPUESTA

Si

No

Sin detalles

Descartar

Seleccione varios amigos manteniendo pulsada la tecla "Control". Si la respuesta es "Sin detalles" descarte a los amigos que tengan información sobre la pregunta.

Figura 108. Realización de la prueba PR-010.



Identificador	PR-011
Fecha	3 de Septiembre de 2011
Responsable	Programador
Descripción	Usuario realiza pregunta completar X.
Resultado	Se trata de una prueba genérica que deberá ser repetida para las preguntas indicadas en el proceso de prueba. En el caso de la figura 109 se realiza la pregunta 4 y se comprueba que la pregunta esté bien formulada observando la respuesta que da Com a ésta. Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 135. Plantilla prueba PR-011.

Cuando haya terminado de descartar pulse turno.

04. ¿Nació en la comunidad de ...?

Madrid

RESPUESTA

Para cambiar de turno una vez que haya terminado de descartar, pulse el botón "Turno".

Figura 109. Realización de la prueba PR-011.




Identificador	PR-012
Fecha	4 de Septiembre de 2011
Responsable	Programador
Descripción	Com realiza pregunta completar X.
Resultado	Se trata de una prueba genérica que deberá ser repetida para las preguntas indicadas en la prueba. En el caso de la figura 110 se realiza la pregunta 16 y se comprueba que la pregunta guarda relación con un amigo no descartado del tablero de Com y que no ha sido formulada con anterioridad. Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 136. Plantilla prueba PR-012.

Ver detalles del objetivo

Seleccione un amigo(a) ...



Alvaro Mejia Sevilla

Estado sentimental: Tiene una relación

Sexo: Hombre

Lugar de nacimiento: Madrid, Spain

Fecha de nacimiento : 03/28/1986

Responda a la pregunta sobre el amigo objetivo.

16. ¿Nació en el mes 01?

RESPONDER

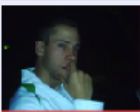
El amigo objetivo puede verlo en la parte inferior izquierda de la pantalla. Si no tiene especificada la información para dicha pregunta pulse el botón "Sin detalles".

Figura 110. Realización de la prueba PR-012.

Identificador	PR-013
Fecha	4 de Septiembre de 2011
Responsable	Programador
Descripción	Com realiza pregunta completar no X.
Resultado	Se trata de una prueba genérica que deberá ser repetida para las preguntas indicadas en la prueba. En el caso de la figura 111 se ha realizado la pregunta 6 y se comprueba que la pregunta guarda relación con un amigo no descartado del tablero de Com y que no ha sido formulada con anterioridad. Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 137. Plantilla prueba PR-013.

Ver detalles del objetivo
Seleccione un amigo(a) ...



Alvaro Mejia Sevilla
Estado sentimental: Tiene una relación
Sexo: Hombre
Lugar de nacimiento: Madrid, Spain
Fecha de nacimiento : 03/28/1988

Responda a la pregunta sobre el amigo objetivo.

06. ¿Su estado sentimental es soltero(a)?

RESPONDER

El amigo objetivo puede verlo en la parte inferior izquierda de la pantalla. Si no tiene especificada la información para dicha pregunta pulse el botón "Sin detalles".

Figura 111. Realización de la prueba PR-013.

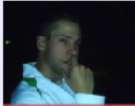


Identificador	PR-014
Fecha	4 de Septiembre de 2011
Responsable	Programador
Descripción	Respuesta Usuario Y.
Resultado	Se trata de una prueba genérica que deberá ser repetida para las respuestas indicadas en la prueba. En el caso de la figura 112 se ha utilizado la respuesta <i>No</i> . Se comprobará que la respuesta ha llegado bien al oponente observando si los descartes realizados concuerdan con la respuesta. Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 138. Plantilla prueba PR-014.

Ver detalles del objetivo

Seleccione un amigo(a) ...



Alvaro Mejia Sevilla

Estado sentimental: Tiene una relación

Sexo: Hombre

Lugar de nacimiento: Madrid, Spain

Fecha de nacimiento : 03/28/1986

Responda a la pregunta sobre el amigo objetivo.

06. ¿Su estado sentimental es soltero(a)?

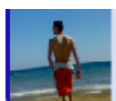
RESPONDER

El amigo objetivo puede verlo en la parte inferior izquierda de la pantalla. Si no tiene especificada la información para dicha pregunta pulse el botón "Sin detalles".


Figura 112. Realización de la prueba PR-014.

Identificador	PR-015
Fecha	5 de Septiembre de 2011
Responsable	Programador
Descripción	Respuesta Com Y.
Resultado	Se trata de una prueba genérica que deberá ser repetida para las respuestas indicadas en la prueba. En el caso de la figura 113 se utiliza la respuesta <i>SI</i> . Se comprobará si la respuesta es la adecuada comparándola con la información del amigo objetivo que usuario ha de descubrir y que se encuentra en la <i>interfazCom</i> . Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

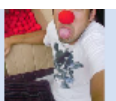
Tabla 139. Plantilla prueba PR-015.



Madrid, Spain
Soltero(a)
Hombre

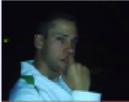


Soltero(a)
Mujer



Castilla y Leon, Spain
Soltero(a)
Hombre

[Ver detalles del objetivo](#)
[Seleccione un amigo\(a\) ...](#)



Alvaro Mejia Sevilla
Estado sentimental: Tiene una relación
Sexo: Hombre
Lugar de nacimiento: Madrid, Spain
Fecha de nacimiento : 03/28/1986

01. ¿Es hombre?

RESPUESTA

Seleccione varios amigos manteniendo pulsada la tecla "Control". Si la respuesta es "Sin detalles" descarte a los amigos que tengan información sobre la pregunta.

Figura 113. Realización de la prueba PR-015.

Identificador	PR-016
Fecha	5 de Septiembre de 2011
Responsable	Programador
Descripción	Descartar amigos/as tablero Usuario.
Resultado	Se comprobará la funcionalidad del botón <i>Descartar</i> , figura 114. Una vez seleccionados todos aquellos amigos que han de ser descartados, se pulsará el botón <i>Descartar</i> , y automáticamente quedarán marcados como descartados. Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 140. Plantilla prueba PR-016.




Figura 114. Realización de la prueba PR-016.

Identificador	PR-017
Fecha	5 de Septiembre de 2011
Responsable	Programador
Descripción	Descartar amigos/as tablero Com
Resultado	<p>Se comprobará que Com es capaz de descartar correctamente a los amigos/as precisos de su tablero en función de la respuesta dada por usuario a su pregunta anteriormente realizada. En la figura 115 se realiza la pregunta 06. <i>¿Su estado sentimental es soltero(a)?</i>, se responderá <i>Sin detalles</i> ya que, como se puede observar, el amigo objetivo no tiene especificado su estado sentimental.</p> <p>El resultado de descarte se observa en la figura 116, en la que se observa que en el tablero de Com han sido descartados todos aquellos amigos que tienen especificado su estado sentimental de manera que se cumple con la respuesta dada por Usuario. Prueba superada satisfactoriamente.</p>
Descripción de los fallos	Ninguno.

Tabla 141. Plantilla prueba PR-017.

Ver detalles del objetivo
Selecione un amigo(a) ...



Alvaro Estuñiga Marcos

Sexo: Hombre

Lugar de nacimiento: Madrid, Spain

Fecha de nacimiento : 05/29/1985

Responda a la pregunta sobre el amigo objetivo.

06. ¿Su estado sentimental es soltero(a)?

RESPONDER

El amigo objetivo puede verlo en la parte inferior izquierda de la pantalla. Si no tiene especificada la información para dicha pregunta pulse el botón "Sin detalles".

Figura 115. Realización de la prueba PR-017. Pregunta lanzada por Com.



Figura 116. Realización de la prueba PR-017. Situación del tablero de Com después de la respuesta.

Identificador	PR-018
Fecha	6 de Septiembre de 2011
Responsable	Programador
Descripción	Resolver Usuario.
Resultado	<p>Se comprobará la funcionalidad del botón <i>Resolver</i>. Usuario ya ha descartado todas las posibles opciones y decide resolver seleccionando el amigo que cree ser su objetivo y haciendo clic sobre <i>Resolver</i>. Si el resultado es correcto la interfaz muestra que ha ganado, figura 117.</p> <p>Se comprobará el resultado acudiendo a la interfaz de Com y haciendo clic sobre el botón <i>Amigo/a objetivo</i>. <i>Pulse el botón para ver detalles</i>, figura 118, de manera que se comparará con el resultado propuesto por usuario, teniendo que coincidir en todo caso. Prueba superada satisfactoriamente.</p>
Descripción de los fallos	Ninguno.

Tabla 142. Plantilla prueba PR-018.



Figura 117. Realización de la prueba PR-018. Interfaz Usuario Resolver.

Amigo/a objetivo. Pulse el botón para ver detalles.



Ignacio Garcia Moreno
 Estado sentimental: Soltero(a)
 Sexo: Hombre
 Lugar de nacimiento: Madrid, Spain
 Fecha de nacimiento : 04/22/1988

Seleccione un amigo(a) para ver detalles

Estudios:
 Centro: av. de los toreros
 Residencia: Madrid, Spain
 Localidad de nacimiento: Madrid, Spain

Figura 118. Realización de la prueba PR-018. InterfazCom, objetivo Usuario.

Identificador	PR-019
Fecha	6 de Septiembre de 2011
Responsable	Programador
Descripción	Resolver Com.
Resultado	<p>Se comprobará en esta prueba la capacidad de Com para resolver y ganar una partida. En el caso de la figura 119 se muestra que el amigo a descubrir es <i>Solo Méndez</i>. En el tablero de Com, deberían de estar descartados todos los amigos/as excepto el amigo ya comentado, figura 120.</p> <p>Cuando la <i>interfazUsuario</i> indique que Com ha ganado la partida, se irá a la <i>interfazCom</i>, de manera que se comprobará que en su tablero sólo queda por descartar un amigo que corresponde con el amigo objetivo que tenía que descubrir y que Usuario podía observar en su vista. Como se observa en la figura 120 es <i>Solo Méndez</i> el único que queda sin descartar. Prueba superada satisfactoriamente.</p>
Descripción de los fallos	Ninguno.

Tabla 143. Plantilla prueba PR-019.



Figura 119. Realización de la prueba PR-019. InterfazUsuario, Com gana la partida.



Figura 120. Realización de la prueba PR-019. InterfazCom, Comprobando tablero.



Identificador	PR-020
Fecha	7 de Septiembre de 2011
Responsable	Programador
Descripción	Salir del juego.
Resultado	Se comprobará la funcionalidad del botón <i>Salir</i> asegurando que se produce la desconexión de <i>Facebook</i> . Una vez se ha hecho clic sobre dicho botón se saldrá del juego, y si se hace clic sobre cualquier elemento del <i>iFrame</i> de <i>Facebook</i> aparecerá un mensaje indicando que se está desconectado, figura 121. Prueba superada satisfactoriamente.
Descripción de los fallos	Ninguno.

Tabla 144. Plantilla prueba PR-020.

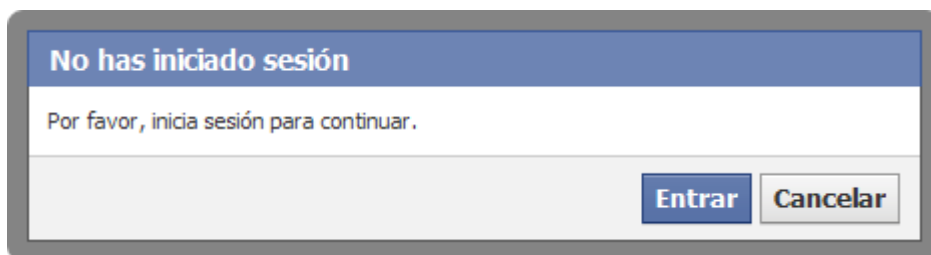


Figura 121. Realización de la prueba PR-020. Desconexión de *Facebook*.

Identificador	PR-021
Fecha	7 de Septiembre de 2011
Responsable	Programador
Descripción	Usuario cede turno.
Resultado	<p>Se comprobará la funcionalidad del botón <i>Turno</i>, figura 122. De manera que cuando se haga clic sobre dicho botón, Com empiece su turno. Una vez que Usuario haya terminado de descartar, usuario hace clic sobre el botón en cuestión.</p> <p>Se le cederá el turno a Com y este en función de cómo haya avanzado en el juego, resolverá o hará una pregunta a usuario. En este ejemplo, figura 123, realizará otra pregunta. Prueba superada satisfactoriamente.</p>
Descripción de los fallos	Ninguno.

Tabla 145. Plantilla prueba PR-021.

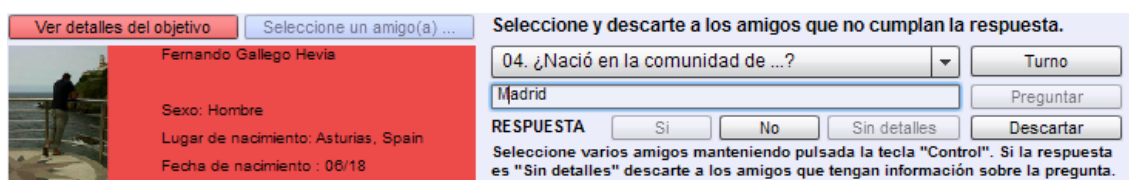


Figura 122. Realización de la prueba PR-021. Descarte Realizado, ceder turno.

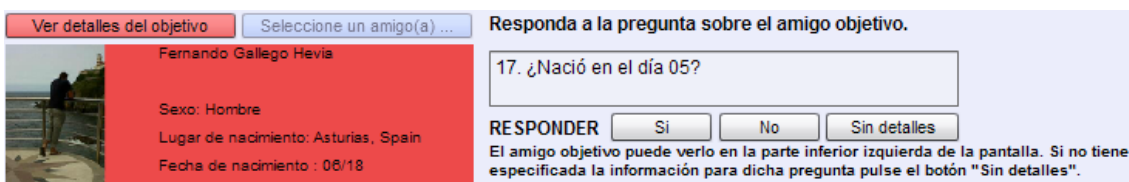


Figura 123. Realización de la prueba PR-021. Turno cedido, pregunta Com.

9.2 Test de valoración

Se ha realizado un test a un reducido grupo de personas para conocer sus opiniones con respecto a la aplicación desarrollada. La selección de estas personas se ha realizado detenidamente, escogiendo distintos rangos de edad, sexo, experiencia en internet (redes sociales) y ocupación profesional.

La prueba consistirá en jugar 3 partidas en la aplicación con el fin de que los sujetos sometidos al test tengan suficiente información para poder sacar conclusiones y realizar sus valoraciones.

Las opiniones de éstos serán recogidas en unos test en los que valorarán las distintas opciones entre 1 y 5. Las opiniones versarán sobre:

- **Interfaz:** si ésta es atractiva o no para los usuarios.
- **Jugabilidad:** los participantes del test indicarán si han disfrutado y, por tanto, si volverían a jugar.
- **Intuitivo:** si les ha parecido sencillo el desarrollo de la partida.
- **Valoración global:** valoración final de la aplicación por parte del sujeto

Finalmente, se realizará un pequeño estudio para sacar conclusiones que permitan realizar una valoración utilizando un punto de vista externo al proyecto.

Los resultados de los test se muestran en las tablas 125-134.

Prueba Sujeto 1	
Sexo	Varón
Edad	61
Experiencia internet	Media
Ocupación profesional	Prejubilado
TEST	
Interfaz	4
Jugabilidad	4
Intuitivo	3
VALORACIÓN GLOBAL	4

Tabla 146. Test de valoración. Sujeto 1.



Prueba Sujeto 2	
Sexo	Mujer
Edad	62
Experiencia internet	Baja
Ocupación profesional	Sus labores
TEST	
Interfaz	3
Jugabilidad	3
Intuitivo	2
VALORACIÓN GLOBAL	3

Tabla 147. Test de valoración. Sujeto 2.

Prueba Sujeto 3	
Sexo	Mujer
Edad	35
Experiencia internet	Media
Ocupación profesional	Abogada
TEST	
Interfaz	4
Jugabilidad	5
Intuitivo	3
VALORACIÓN GLOBAL	4

Tabla 148. Test de valoración. Sujeto 3.

Prueba Sujeto 4	
Sexo	Mujer
Edad	31
Experiencia internet	Alta
Ocupación profesional	Informática
TEST	
Interfaz	5
Jugabilidad	4
Intuitivo	4
VALORACIÓN GLOBAL	4

Tabla 149. Test de valoración. Sujeto 4.



Prueba Sujeto 5	
Sexo	Mujer
Edad	28
Experiencia internet	Media
Ocupación profesional	Profesora
TEST	
Interfaz	4
Jugabilidad	5
Intuitivo	3
VALORACIÓN GLOBAL	4

Tabla 150. Test de valoración. Sujeto 5.

Prueba Sujeto 6	
Sexo	Varón
Edad	25
Experiencia internet	Alta
Ocupación profesional	Informático
TEST	
Interfaz	5
Jugabilidad	3
Intuitivo	4
VALORACIÓN GLOBAL	4

Tabla 151. Test de valoración. Sujeto 6.

Prueba Sujeto 7	
Sexo	Varón
Edad	26
Experiencia internet	Media
Ocupación profesional	Perito Mecánico
TEST	
Interfaz	4
Jugabilidad	3
Intuitivo	4
VALORACIÓN GLOBAL	4

Tabla 152. Test de valoración. Sujeto 7.



Prueba Sujeto 8	
Sexo	Varón
Edad	16
Experiencia internet	Media
Ocupación profesional	Estudiante
TEST	
Interfaz	5
Jugabilidad	3
Intuitivo	5
VALORACIÓN GLOBAL	4

Tabla 153. Test de valoración. Sujeto 8.

Prueba Sujeto 9	
Sexo	Varón
Edad	25
Experiencia internet	Baja
Ocupación profesional	Comercial
TEST	
Interfaz	3
Jugabilidad	3
Intuitivo	3
VALORACIÓN GLOBAL	3

Tabla 154. Test de valoración. Sujeto 9.

Prueba Sujeto 10	
Sexo	Mujer
Edad	27
Experiencia internet	Media
Ocupación profesional	Comercial
TEST	
Interfaz	5
Jugabilidad	5
Intuitivo	3
VALORACIÓN GLOBAL	4

Tabla 155. Test de valoración. Sujeto 10.

9.2.1 Conclusiones test de valoración

La muestra sobre la que se ha hecho el estudio era homogénea en cuanto a hombres y mujeres. A continuación se muestra un estudio de la media total y de las medias por sexo.

Media total	
Interfaz	4,2
Jugabilidad	3,7
Intuitivo	3,4
VALORACIÓN GLOBAL	3,8

Tabla 156. Media total.

Medias Totales Mujeres	
Interfaz	4,2
Jugabilidad	4,4
Intuitivo	3,0
VALORACIÓN GLOBAL	3,8

Tabla 157. Media total Mujeres.

Medias Totales Varones	
Interfaz	4,2
Jugabilidad	3,2
Intuitivo	3,8
VALORACIÓN GLOBAL	3,8

Tabla 158. Media total Varones.

Estudiando las tablas 135-137, se puede ver que la media global sale igual tanto para hombres como para mujeres. Esto indica, que gusta a ambos por igual. No obstante los resultados no coinciden en todos los aspectos.

En otro atributo en el que coinciden es en la interfaz. A ambos grupos les ha gustado la vistosidad de la aplicación, destacando colores y colocación de los elementos del programa.

Por otro lado, mientras los hombres destacan la intuitividad de la aplicación, a las mujeres les cuesta más hacerse a los numerosos botones que presenta ésta, y se les hace un poco complicada la primera partida. Según avanzan los turnos, se hacen con los controles alcanzando más satisfacción en las partidas que los varones.

Éstos consideran que la jugabilidad está un poco limitada, ya que, el sólo poder enfrentarse con la máquina lo ven escaso y les gustaría enfrentarse con otros oponentes



o amigos. Las mujeres también valoran esa opción, pero disfrutan más de las partidas contra la Com que los varones.

A nivel individual, se observa que las notas bajan ligeramente a medida que sube la edad y notablemente en función de la carencia de experiencia en el manejo de internet, dos últimos sujetos del gráfico representado en la tabla 124. Aquellos que no realizan un alto uso de las redes experimentan un primer contacto complicado a la hora de jugar con la aplicación aunque con el paso de los turnos y las partidas, se les hace todo más sencillo.

Finalmente, destacar que la aplicación gusta a todos más o menos por igual, destacando que la gente con más conocimientos en informática valora más la aplicación que aquellos que no hacen un gran uso de las nuevas tecnologías, tercer sujeto del gráfico de la figura 124. Además, sería conveniente realizar un estudio para intentar aumentar la intuitividad de la aplicación, para mejorar la experiencia de algunos usuarios en las primeras partidas. También la inclusión de un modo de juego en el que se pudiera retar a amigos de la red social ampliaría la experiencia que se puede obtener de la aplicación.

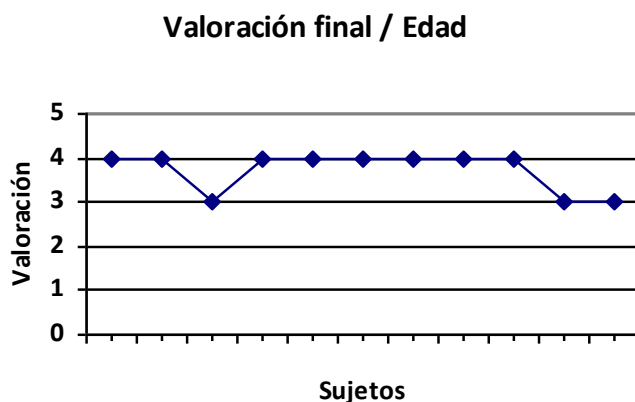


Figura 124. Gráfico Valoración final / Edad.



10. Conclusiones

A la hora de realizar el presente proyecto se pretendía alcanzar una serie de objetivos que se marcaron al inicio, consiguiendo llegar a todos ellos de manera satisfactoria.

En primer lugar, se ha creado una aplicación que proporciona la funcionalidad deseada, es decir, la interacción con las Bases de Datos que *Facebook* pone a disposición de los desarrolladores y el posterior uso de estos datos.

Se ha llevado a cabo el proyecto siguiendo una adaptación de la metodología *Métrica Versión 3*, permitiendo desarrollar un software con cierto nivel de calidad, así como su documentación.

Se ha seguido la planificación establecida al principio del proyecto cumpliendo los plazos de entrega establecidos al comienzo.

Se han puesto en práctica los conocimientos adquiridos a lo largo de la carrera, sobre todo en el uso de metodologías, programación orientada a objetos, servidores, páginas web mediante *HTML*, técnicas de inteligencia artificial y gestión de proyectos software. Además, se han adquirido nuevos conocimientos gracias al aprendizaje de las tecnologías *MXML* y *ActionScript* utilizadas en el desarrollo del sistema, e incluso, adquiriendo experiencia a la hora de desarrollar aplicaciones integradas en *Facebook*.

Finalmente, se ha logrado alcanzar el objetivo de desarrollar un proyecto software completo, adquiriendo conocimientos acerca de los roles presentes en este tipo de proyectos y comprobando la dificultad en lo referente a esfuerzo, dedicación y coste de llevar a cabo un proyecto de forma correcta.



11. Líneas Futuras

Después de la consecución de este proyecto se podrían ampliar las capacidades y posibilidades de la aplicación. La idea principal sería modificar el software de manera que el juego no sólo enfrentase a un usuario con una IA, sino que dos personas, con cuentas en *Facebook*, pudieran competir con sus amigos correspondientes al *¿Quién es quién?*.

Esta tarea no sería muy complicada. Consistiría en la creación de un chat por turnos, de manera que se sincronizase con la aplicación, con el que los usuarios pudieran formular sus preguntas y poder responderlas fácilmente. Esto aumentaría hasta cotas muy altas la jugabilidad de la aplicación, ya que las preguntas no estarían limitadas y podrían ampliarse para abarcar, incluso, aspectos relativos a la foto del amigo o amiga que forma parte del tablero.

También sería interesante incluir una base de datos conectada a la aplicación, de tal forma que almacenase los enfrentamientos que han tenido lugar. Así podrían crearse clasificaciones sobre usuarios con más victorias, de tal forma que pudieran ser retados por otros usuarios de *Facebook* intentando crear adicción al juego.

Finalmente, contar con un servidor propio para poner todo esto en funcionamiento se antoja vital, ya que así se puede realizar un mantenimiento más exhaustivo y controlar la disponibilidad de la aplicación en Internet. Además, este servidor no sólo alojaría la aplicación, sino también las páginas webs con las instrucciones y clasificaciones antes mencionadas.



Lista de acrónimos

- **LaBdA:** Laboratorio de Bases de Datos Avanzadas.
- **CU:** Casos de Uso.
- **RS:** Requisito Software.
- **RU-C:** Requisito de Usuario de Capacidad.
- **RU-R:** Requisito de Usuario de Restricción.
- **MXML:** Multimedia eXtensible Markup Language.
- **XML:** eXtensible Markup Language.
- **RIA:** Rich Internet Applications.
- **API:** Application programming interface.
- **BBS:** Bulletin Board System.
- **AOL:** America OnLine.
- **FQL:** Facebook Query Language.
- **REST:** Representational State Transfer.
- **FBML:** Facebook Markup Language.
- **URL:** Uniform Resource Locator or Universal Resource Locator.
- **CSS:** Cascading Style Sheets.
- **AFB4:** Adobe Flash Builder 4.
- **IA:** Inteligencia Artificial.
- **RBS:** Resource Breakdown Structure.
- **WBS:** Word Breakdown Structure.
- **IEEE:** Institute of Electrical and Electronics Engineers.
- **UML:** Unified Modeling Language.
- **JSON:** JavaScript Object Notation.



Bibliografía

- Adobe Developer. (2009). Adobe Flex Developer Center. Adobe Systems. <<http://www.adobe.com/devnet/flex.html>>. Accessed 7 Marzo 2011.
- Adobe Developer Connection. (2009) Flex and Facebook. Adobe Systems. <<http://www.adobe.com/devnet/facebook/>>. Accessed 21 Febrero 2011.
- Facebook Developers. (2007) Facebook Developers. Facebook, Inc. <<http://developers.facebook.com/>>. Accessed 21 Febrero 2011.
- Google Code. (2011) Adobe Flash Platform, Facebook-ActionScript-API. Google Inc. <<http://code.google.com/p/facebook-actionscript-api/>>. Accessed 7 Marzo 2011
- Google Code. (2010) Google Code. Google Inc. <<http://code.google.com/intl/es/>>. Accessed 4 Marzo 2011.
- Jeffrey Van Camp. (2011) The history of social networking before Facebook and Twitter. Digital Trends. <<http://www.digitaltrends.com/computing/a-history-of-social-networking-before-facebook-and-twitter/>>. Accessed 10 Febrero 2011.
- Karl Hodge. (2008) Revealed: 20 Facebook apps that don't suck. Future Publishing Limited. <http://www.techradar.com/news/internet/web/revealed-20-facebook-apps-that-don-t-suck-486818?artc_pg=3>. Accessed 11 Febrero 2011.
- Marketing Capacitación (2011) La verdad sobre los social games en Facebook. Marketing Capacitación. < http://www.marketingcapacitacion.com/index.php?option=com_content&view=article&id=70%3Ala-verdad-sobre-los-social-games-en-facebook&catid=39%3Asocial-media&Itemid=58>. Accessed 08 Agosto 2011.
- Mario Iván Cruz (2010) Calidad Del Software en el Ciclo de Vida. ITESCAM.
- PC World. (2009) 10 aplicaciones para sacarle partido a Facebook. IDG Communications. <<http://www.idg.es/pcworld/10-aplicaciones-para-sacarle-partido-a-Facebook/art193644.htm>>. Accessed 11 Febrero 2011.
- Portal de Administración Electrónica (2011) Métrica 3. Portal de Administración Electrónica. Ministerio de Política Territorial y Administración Pública. <http://administracionelectronica.gob.es/?_nfpb=true&_pageLabel=P60085901274201580632&langPae=es>. Accessed 9 Febrero 2011.
- Wikipedia (2011) Caso de uso. Fundación Wikipedia Inc. <http://es.wikipedia.org/wiki/Caso_de_uso> Accessed 25 Marzo 2011.
- Wikipedia (2011) Estructura de descomposición del trabajo. Fundación Wikipedia Inc. <http://es.wikipedia.org/wiki/Estructura_de_descomposici%C3%B3n_del_trabajo> Accessed 1 Abril 2011.



Proyecto Fin de Carrera

Francisco Ruiz Gualda - Creación de la aplicación “¿Quién es quién?” para Facebook

- Wikipedia (2011) RBS. Fundación Wikipedia Inc.
<http://en.wikipedia.org/wiki/Resource_Breakdown_Structure> Accessed 1 Abril 2011

Anexo A: Manual de la Aplicación *PFC - ¿Quién es?*

A continuación se proporciona una guía para el correcto desarrollo del juego, de manera que el usuario tenga una partida fluida desde el comienzo. Este manual describe al usuario el funcionamiento básico de la aplicación.

1 Iniciar juego

Como requerimiento imprescindible para poder jugar a la aplicación *PFC - ¿Quién es?* el usuario ha de tener una cuenta registrada en *Facebook* con al menos 15 amigos agregados.

Para dar comienzo al juego el usuario deberá abrir su navegador y dirigirse a alguna de las siguientes direcciones:

- <https://apps.facebook.com/pfc-quienes/>: permitirá al usuario jugar bajo el *iFrame* de *Facebook*.
- <http://pfc.comuf.com/>: la aplicación se ejecuta en una dirección web que sólo contiene la aplicación.

Una vez que el navegador se haya dirigido a la url correspondiente, automáticamente éste se redirigirá a otra url, figura 125, que permitirá al usuario iniciar sesión para utilizar su cuenta de *Facebook* con *PFC - ¿Quién es?*.

Figura 125. Iniciar sesión *Facebook*.

Una vez que se haya logrado conectar con la red social *Facebook*, el navegador nuevamente le redirigirá a la web anteriormente introducida. Será en ese momento cuando se cargará la aplicación. Una vez que haya terminado, permitirá al usuario pulsar el botón *Empezar juego* dando lugar así al comienzo del juego.

2 Desarrollo de la partida

Se trata de un juego similar al juego de mesa en el que se tiene que adivinar, mediante una sucesión de preguntas y respuestas, que amigo/a de entre todos los que se muestran en el tablero tiene el adversario y viceversa.

Reglas:

1. Ganará aquel que acierte antes el objetivo.
2. Si no se resuelve correctamente, el usuario empezará de cero, es decir, en su tablero se activarán otra vez todos los amigos/as y le corresponderá un nuevo objetivo.
3. Se han de contestar a todas las preguntas con sinceridad.

Instrucciones:

1. Pulsar el botón *Empezar juego*, figura 126.



Figura 126. Empezar juego.

2. Automáticamente se podrá observar el tablero cargado, así como el amigo/a objetivo que tiene que adivinar el oponente. Esta interfaz se muestra en la figura 127 y contiene los siguientes componentes:



Figura 127. Vista general de la aplicación

1. *Tablero*. De entre todos estos amigos se ha de descubrir el objetivo.
2. *Amigo/a objetivo del oponente*. Es el amigo que ha de adivinar el adversario.
3. *Texto de ayuda*. Dará indicaciones sobre lo que se debe hacer en todo momento.
4. *Ir a vista Com*. Este botón permitirá ver el tablero del adversario. Se podrán comprobar sus avances, observar los detalles de los amigos que componen su tablero y el amigo/a objetivo que se tiene que descubrir. Esto último no es muy recomendable si el juego no ha finalizado, ya que perdería sentido la partida.
5. *Ver detalles del objetivo*. Se podrán observar más datos del objetivo del adversario sobre los que realizar preguntas.
6. *Ver detalles de amigo/a*. Pinchando sobre un amigo del tablero de Usuario y posteriormente pulsando dicho botón, se observarán más datos sobre este amigo/a que aparecerán encima del amigo/a objetivo del adversario (segundo punto de la lista). Pinchando sobre otro amigo del tablero, se volverá a ver dicho amigo/a objetivo.
7. *Controles del juego*. En este apartado estarán los controles que permitirán ir desarrollando la partida.

Ahora se verá en mayor profundidad este último punto.

Controles del juego:

1. Al principio de cada apartado se podrá resolver o preguntar, figura 128. Para preguntar se elegirá una de las 26 cuestiones que están a disposición del usuario. Si la pregunta es del tipo *¿Estudió en el centro ...?* se ha de completar rellenando la barra que hay justamente debajo con la información que se desee. Una vez hecho esto se pulsará el botón *Preguntar*.

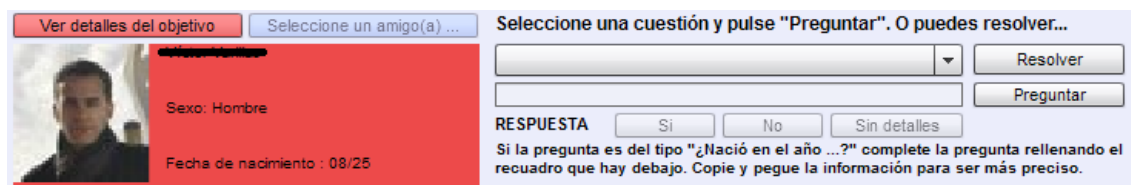


Figura 128. Controles del juego. Pregunta.

2. Automáticamente el adversario contestará a la pregunta activándose el botón de *Respuesta* correspondiente. En este momento se harán visibles los botones *Descartar* y *Turno*, figura 129. Para descartar aquellos amigos/as del tablero que no cumplan la condición, por lo que éstos no podrían ser el objetivo a descubrir por usuario, se seleccionará todos utilizando la tecla *control* y pinchando sobre estos con el ratón. Una vez hecho esto se hará clic sobre *Descartar* y todos los amigos seleccionados quedarán desechados. Si la respuesta fuese *Sin detalles* indicaría que el objetivo no tiene datos sobre dicha pregunta, y habría que descartar todos los amigos/as que tengan detalles acerca de esa cuestión.

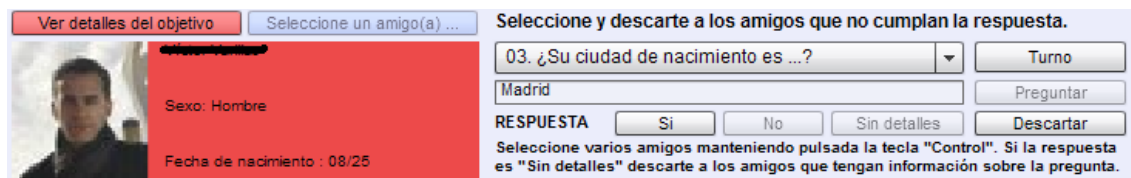


Figura 129. Controles del juego. Respuesta de oponente a pregunta de Usuario. Descartar.

3. Se pulsará *Turno* cuando se haya acabado de descartar amigos/as. El oponente hará su pregunta, usuario deberá contestar sinceramente presionando los botones de *Responder* en función de los datos del objetivo que aparecen en el recuadro rojo, figura 130. Si no hay datos sobre dicha pregunta se hará clic sobre el botón *Sin detalles*.

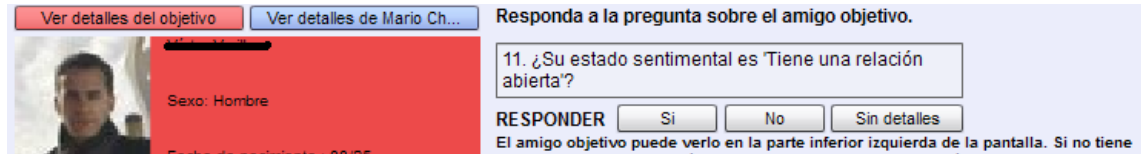


Figura 130. Controles del juego. Respuesta de usuario a pregunta del oponente.

4. Se volverá a conceder el turno a Usuario, volviendo a elegir entre hacer una pregunta o resolver. En este turno se hará clic sobre *Resolver*. En este caso se deberá seleccionar el amigo/a que se cree es la solución y volver a pinchar sobre el botón *Resolver* para averiguar si se ha ganado o si, por el contrario, se ha de empezar de cero, figura 131. Se puede pinchar sobre *Volver* si se desea mejor hacer una pregunta.

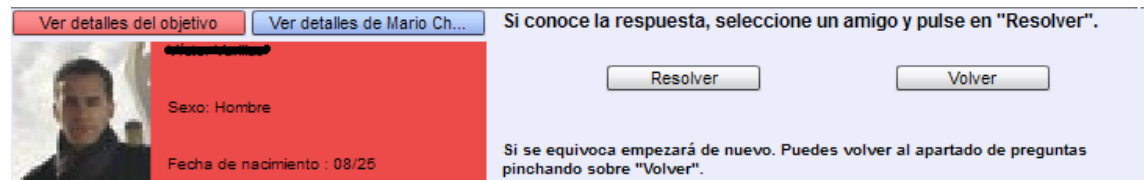


Figura 131. Controles del juego. Resolver.

Anotaciones

La fecha de nacimiento de los amigos/as tiene formato americano, es decir, *mm/dd/yy*. Es decir, tienen cambiado el orden de apariencia de días y meses con respecto al formato español.